עלון קבוצת עניין אחזור טקסט - SIGTRS

יוני 2004 כרך י' מספר 2

כ בכסלו, תשס״ד
1 ביוני, 2004
2004-6-T

חדשות מקבוצת העניין - SIGTRS

שלום לכולם!

אנו נפגשים שוב בעלון חדשות נוסף. זו הזדמנות טובה להודות לסימה לוי משע״ם על התמיכה המתמשכת בצדדים הטכניים של הקבוצה ועל עבודתה המסורה בקשר עם משתתפי הקבוצה.

ראוי לציין שרוב הנושאים הנידונים בקבוצה נקבעים ע״י החברים עצמם בדפי המשוב בסיום המפגשים. אתם מוזמנים להמשיך להציע נושאים לדיון וכמובן להציע את עצמכם להרצאה.

1. <u>קשר</u>

הקבוצה מקימת קשר עם חבריה בשתי דרכים:

- קבוצת תפוצה אלקטרונית (Mailing-list) המאפשרת בצורה חופשית לכל אדם להרשם לקבוצה או למחוק עצמו ממנה (הדרכה כיצד להצטרף או לעדכן כתובת מופיעה באתר תחת התפריט ״רשימת תפוצה של הקבוצה״).

- דואר פיזי (דף הרישום אליו נמצא באתר הקבוצה).

מומלץ למי שרוצה להיות חבר מלא ולקבל את החומר להרשם באתר הן כחבר והן ברשימת התפוצה האלקטרונית. למי שתהליך הרישום מעייף - ניתן לשלוח לי דוא״ל עם פרטיכם האלקטרוניים והפיזיים ואטפל בעניין עבורכם.

באתר הקבוצה מפה שנועדה להקל על ההגעה של החברים מחוץ לעיר. הדפיסו אותה לפני היציאה.
נסיעה טובה!

2. <u>מפגשים</u>

מאז הוצאת הגליון האחרון (כרך י׳ חוברת מס. 1) בינואר 2004 נפגשה הקבוצה פעמיים. המפגש הראשון התקיים בינואר 2004. במפגש נשמעו ההרצאות:

א. השוואה בין מימושים שונים של מורפולוגיה עברית ביישומי אחזור מידע טקסטואלי ע״י אפרים מרגלית מהאוניברסיטה הפתוחה.
בהרצאה נסקרו מושגי יסוד במורפולוגיה וכן שיטות שונות למימושם.
הוצגה השוואה בין השיטות השונות וכן יתרונות וחסרונות לכל שיטה.

ב. אחזור מידע ומורפולוגיה של השפה הערבית ע״י דרור קמיר מח׳ מלינגו.
בהרצאה הוצגו מושגים למורפולוגיה תוך דגש על דוגמאות מהשפה הערבית.
הוצגו דרכי התמודדות של מורפיקס עם השפה הערבית ודרכי המימוש וכן הוצגו פתרונות לסוגי בעיות נפוצות באחזור מידע של מסמכים בערבית כולל התיחסות לדיאלקטים השונים של השפה הערבית בארצות ערב השונות.

המפגש השני התקיים באפריל 2004. במפגש נשמעו ההרצאות:

א. איתור נושא מסמך בצורה אוטומטית תוך שימוש במילים נפוצות, ע״י עפר דרורי משע״ם.
בהרצאה הוצגו ממצאי מחקר שבדק את היכולת לזיהוי נושא מסמכים בצורה אוטומטית תוך שימוש בכלי המבוסס על ניתוח הטקסט והמילים הנפוצות שבו. ממצאי המחקר הראה שניתן לאתר בצורה אוטומטית את נושא המסמך עד כדי 90% תלוי בסוג החומר.

ב. ייצור אוטומטי של תיזאורי ומילונים דו לשוניים ע״י איריס ארד.
בהרצאה הוצגו שיטות ליצור אוטומטי של מילונים דו לשוניים ככלי עזר לבלשנים וכן הוצגו כלים העושים שימוש בשיטות אלו.

כמו כן בעלון מאמר נוסף:

- עבודה בנושא עידון תהליכי חיפוש והצגתם למשתמש שחוברה ע״י זיו סלייטר ורחל בן עזרא במסגרת לימודיהם במכללה האקדמית הדסה לתואר ראשון במדעי המחשב. העבודה בוצעה בהנחייתו של ד״ר עפר דרורי.

קריאה מהנה.

3. **<u>עבודות</u>**

אני מנצל סעיף זה בבקשה לקבלת עבודות או דוחות שונים שבוצעו במסגרות שונות (עבודה, אוניברסיטה וכו׳) העוסקות בנושאי הענין של הקבוצה לצורך פרסומם וכמובן אני מעודד את כל אחד ואחת מכן לשלוח מאמר מפרי עטו בנושאי הקבוצה.

4. **<u>חסות</u>**

אנו מודים לנותני החסות של הקבוצה:
שע״ם (שרות עיבודים ממוכנים)
מרכז החישובים של האוניברסיטה העברית בירושלים.

5. **<u>כללי</u>**

קבוצת הענין "אחזור טקסט" (SIGTRS) מהווה פורום לאנשי מקצוע העוסקים בתחום אחזור הטקסט וטכנולוגיות קשורות כדוגמת Hypertext.
אנשי המקצוע הם מפתחים (מנתחי מערכות ותוכניתנים) ו/או משתמשים.
הקבוצה נפגשת מספר פעמים בשנה להרצאות ולהחלפת רעיונות.
הצעות להרצאות, רעיונות או חומר כתוב אחר ניתן לשלוח ליו"ר הקבוצה:

עפר דרורי
שע"ם
רח' פועלי צדק 4 ירושלים
ת.ד. 10414
ירושלים 91103
פקס: 02-5688681
טל: 02-5688439
או בדואר אלקטרוני -offerd@cc.huji.ac.il

ב ב ר כ ה

עפר דרורי
יו"ר הקבוצה

בקרו אותנו באתר הבית של הקבוצה
http://sigtrs.huji.ac.il

השוואה בין מימושים שונים של מורפולוגיה עברית ביישומי אחזור מידע טקסטואלי

אפרים מרגלית

השוואה בין גישות שונות למימוש ישומי אחזור תוך שימוש במורפולוגיה עברית

"**ויהי כל הארץ שפה אחת ודברים אחדים**" (בראשית, י"א א')

נכתב בהנחייתה של הד״ר מירה בלבן

ע״י אפרים מרגלית

1. מבוא

הטיפול בטקסטים מלווה את האנושות מתקופותיה הקדומות ביותר. השפות לבשו ופשטו צורה. מצורת ביטוי של ציורים עברה האנושות לכתיבה באמצעות תוים גרפיים דמויי ציורים עד לכתיבה באמצעות אותיות. כך שהשימוש בטקסט מלווה את ההיסטוריה האנושית אלפי שנים.

העברית אף היא עברה גלגולים רבים. צורת האותיות השתנתה מהלוחות שהתגלו בגזר ועד לגופנים בני ימינו. השפה עצמה חיה ועברה שינויים החל מלשון המקרא, לשון התקופה התלמודית, ימי הביניים ועד לעברית המודרנית.

צורת המידע ששימשה את הדור הראשון של המחשבים היתה טקסטואלית. אולם גם כיום כאשר עולם המחשבים מתמודד עם תמונות, קול ווידיאו, נראה שלא נס ליחו של המידע הטקסטואלי ועדיין הבכורה נותרה למידע זה.

הטקסטים האגורים במחשבים הולידו צרכים לטיפולם, שעבורם פותחו ישומים רבים. הבולטים שבהם הם תמצות מסמכים, מערכות אחזור טקסטואליות, תוכנות ניקוד, תיקוני הקלדה, Text to speech ומילונים ממוחשבים.

פיתוח ישומים אלו נעשה בכיוונים שונים. לדוגמא, Oracle/Context המבצע תמצות מסמכים, TTS של AT&T המשמש להקראת טקסטים ומוצר האחזור הטקסטואלי של Verity.

ישומים דומים שפותחו עבור השפה העברית נדרשו להתמודד עם בעיות יחודיות לשפה. לדוגמא, מוצרים לאחזור מידע טקסטואלי בעברית נדרשו לאחזר עבור פועל את מאות ואלפי צורות ההטיה שלו. לעומת השפה האנגלית שבה הבעיה פשוטה יחסית. הטיפול בצורות הלשוניות של השפה העברית, המורפולוגית עברית, נעשה בכלים הממוחשבים באופנים שונים.

תחום זה נמצא בשילוב המדעים, והחוקרים העוסקים בתחום זה נדרשו להתמצא הן בבלשנות עברית והן במדעי המחשב.

בעבודה זו ננסה לבחון את הצורך בכלים לניתוח מורפולוגי בעברית, את הפתרונות התיאורטיים והישומיים של כלים אלו תוך התמקדות במערכות אחזור טקסטואליות ובדגש על בעיית רב המשמעויות של השפה, העמימות הלקסיקלית.

2. רקע היסטורי

אחת היכולות הבסיסיות שנדרשו ממערכות ניהול מסמכים היא היכולת לאתר מסמכים. היכולת הבסיסית ביותר בתחום זה היתה חיפוש בטקסט חופשי ללא מפתוח מקדים (Free-text scan). יכולת זו ניתן היה לממש כל עוד גודל בסיס הנתונים לאחזור היה קטן יחסית [Dewire, 1994].

השלב הבא היה אחזור טקסט על פי מילות מפתח (Key word indexing). אחזור זה כלל למעשה שני שלבים מקדימים. הראשון, קביעת מילות המפתח והשני בניית רשימות המילים.

שלב המפתוח הוא למעשה תהליך של "תקצור" המידע וקביעת המילים המייצגות מידע זה. בדומה לתהליך שיוך לנושאים שנעשה בספריות. שלב זה נעשה באופן ידני וקביעת המילים נעשתה על פי ההחלטה של המשתמש.

בשלב השני נבנו רשימות המופעים של המילים. הרשימות הם קבצים מהופכים הכוללים כניסה עבור כל מילת מפתח. בכל כניסה מאוחסנים מספרי יחידות המידע הרלוונטיות [Dewire, 1994].

שימוש במילות מפתח נתן אמנם יכולת אחזור ממאגרי מידע טקסטואליים גדולים אך יצר בעיות אחרות.

הראשונה שבהן תלויה בגורם האנושי. היות ותהליך המפתוח הינו ידני, ההחלטה על מילות המפתח תלויה באדם המבצע את המפתוח ובאסוציאציות אותן חושב הוא לנכון לקשור ליחידת המידע.

בנוסף, מסתבר כי לעיתים עלה צורך באחזור מידע שלא נכלל במפתוח המקדים למרות שהינו חלק מן הטקסט.

בעיה נוספת היא כלכלית. קביעת קטגוריות ליחידת מידע נעשית בתהליך הדורש תשומות גבוהות של משאבי כח אדם.

אמנם נעשה נסיון לפתור חלק מן הבעיות באמצעות כלי עזר כמו שימוש בתזאורוסים ובעצי מלות מפתח. אך הבעיות העיקריות נותרו בעינן והובילו לפיתוח של כלים לאחזור טקסטואלי מלא.

אחזור טקסטואלי מלא (Full-text retrieval) בדומה לאחזור על פי מילות מפתח, התבסס אף הוא על הקמת קבצים מהופכים. השוני העיקרי היה שכל מילה בטקסט היא למעשה "מילת מפתח". בפועל, הסתבר שמילים שכיחות כדוגמת: "the", "a", "an", "for" באנגלית או "את", "על", "של" ו-"אל" בעברית מופיעות ברוב יחידות המידע אך כמעט ואינן משתתפות בתהליך האחזור. מילים אלו נקראות מילות רעש (noise words) או מילות עצר (stop words). על מנת שיקוצר תהליך בניית האינדקסים ויקטן הגודל של האינדקסים לא נבנו אינדקסים למילים אלו [Dewire, 1994].

אם דמינו את מילות המפתח לשיוך נושאי בספריה, ניתן לדמות את האחזור מטקסט חופשי לקונקורדנציה.[Witten, 1994]

הבעיות העיקריות עמן התמודדו האקדמיה והתעשיה בראשית ימי ה-Full text, היו שיפור זמן ההקמה של האינדקסים והקטנת היחס בין גודלו של הטקסט הגולמי לנפח האינדקסים הנובעים ממנו.

האלגוריתמים התפתחו בשני כיוונים, דחיסת הטקסטים ודחיסת האינדקסים. אלגוריתמים לדחיסת מידע התפתחו לצרכים רבים. הבולטים שבהם הם קידוד Huffman ו-Ziv-Lempel. השימושים הראשונים של קידוד Huffman, שפותחו עוד בראשית שנות החמישים, הגיעו ליצוג ממוצע של חמש סיביות לתו בודד. בעוד שהכיווץ על פי Ziv-Lempel שפותח בסוף שנות השמונים, הגיע לכדי יצוג של ארבע סיביות לתו יחיד. שיטות כיווץ מתמטיות הגיעו לייצוג ממוצע של זוג סיביות לתו יחיד. [Witten, 1994].

במקביל, התפתחו אלגוריתמים לניהול יעיל של אינדקסים. האינדקסים נשמרים לרוב בקבצי הצבעות מהופכים (Inverted files) או במפות ביטיות (Bit maps). האלגוריתמים הקטינו את נפח האחסון הנדרש לניהול האינדקסים והן את זמן העיבוד הנדרש לפעולות על האינדקסים. [Witten, 1994]

3. הגדרת הבעיה - במילים אחרות מדוע יש צורך במורפולוגיה עברית

בעוד שבאנגלית המילים המופיעות בטקסט דומות לצורתן הבסיסית (הצורה המילונית), הבעיה בטקסטים עבריים סבוכה הרבה יותר.

בפרק זה ננסה לעמוד על ההיבטים היחודיים של השפה העברית.

3.1 הבעיתיות הנובעת מהמאפיינים היחודיים של השפה העברית

א. הוספת מוספיות בתחילת המילה ובסופה.

ש**מר** -> **שמר**-תם, ת-**שמר**-ו.
ילד -> **ילד-**ים, **ילד**-נו

ב. הוספת מוספיות הכוללות שינוי במבנה הבסיס:

קנה -> **קני**-תי, **קנ**-ו.
שמלה -> **שמלת-**י, **שמל**-ות

ג. קושי בזיהוי תחילית הבסיס בשל הוספת מוספיות:

כשירות -> בסיס: כשירות, שירות.
כשיירות -> בסיס: שיירה
כשירייה -> בסיס: יריה

ד. עמימות בשל כתיב חסר ניקוד

מפלגה -> מפלגה, מ-פלגה מ-פלג-ה, מפלג-ה.
משכן -> משכן, מ-שכן, משכ-ן, מש-כן.

ה. ריבוי ייצוגים אורתוגרפיים שונים לאותה מילה (כתיב חסר, מלא ויתיר):

מזודה -> מזוודה, מזודה, מיזודה, מיזוודה.
לווין -> לוין, לווין, לויין, לוויין.

[בנטור, 1992]

3.2 משמעות הבעיתיות בהיבט האחזור

לרוב, הפונקציונליות הנדרשת מאחזור מידע טקסטואלי היא התעלמות מההבדלים בין הצורות השונות של הבסיס. כך אחזור על פי צורת הבסיס "שמלה" צריך לאחזר גם את קידומת השם כמו "השמלה", שינויים בצורה כמו "שמלת", צורות ריבוי "שמלות", נטיות "שמלתי" ושילובים נוספים כ-"ולכששמלותיהן".

למילה במילון עשויות להתפתח למספר רב של צורות. לדוגמא, הטיות הפועל י.צ.ר (רשימה חלקית):

יצר יצרתי יצרת יצרה יצרנו יצרתם יצרתן יצרו יוצר יוצרת יוצרים יוצרות אצור תיצור יצור ניצור תיצורנה ויצר ויצרתי ויצרת ויצרה ויצרנו ויצרתם ויצרתן ויצרו ויוצר ויוצרת ויוצרים ויוצרות ואצור ותיצור ויצור וניצור ותיצורנה שיצר שיצרתי שיצרת שיצרה שיצרנו שיצרתם שיצרתן שיצרו שיוצר יוצרת שיוצרים שיוצרות שאצור שתיצור שיצור שניצור שתיצורנה ושיצר ושיצרתי ושיצרת ושיצרה שיצרנו ושיצרתם ושיצרתן ושיצרו ושיוצר ושיוצרת ושיוצרים ושיוצרות ושאצור ושתיצור ושיצור שניצור ושתיצורנה כשיצר כשיצרתי כשיצרת כשיצרה כשיצרנו כשיצרתם כשיצרתן כשיצרו כשיוצר כשיוצרת כשיוצרים כשיוצרות כשאצור כשתיצור כשיצור כשניצור כשתיצורנה ולכשתיצורנה ומהיוצרים וביוצרות ומהיוצרות ומיוצרותיהם יוצריהן יוצריכם ויוצריו מיוצרינו כשיוצריה משיוצריו יוצרי וביוצרי וביוצריהן יצירה ויצירות ומהיצירות יצירותיו ומיצירותינו יצירת מיצירת ליצירת שיצירת שמיצירת ושמיצירת ולכשמיצירת ולכשמיצירות וליצירות ויצירות ומיצירות וביצירתנו וכיצירות שמיצירותיכם ושמיצירותיהן ...

[מט"ח, 2000]

מתוך כחמשת אלפים שורשים הקיימים בעברית, מתפתחות מאה מיליון צורות חוקיות. מספר הצורות החוקיות לפועל יחיד יכול להגיע עד כדי 22,000.
[Choueka,1978]

נראה אם כן כי לרוב השימושים של אחזור טקסטואלי, נדרשת יכולת שליפה שתתעלם מההבדלים בין הצורות השונות של מילה בסיסית.

בעיית העמימות היא משמעותית. היות ורק 45% מהמילים העבריות הינן חד משמעיות.
[Choueka, 1978]

לסיכום, הבעיתיות בישומי עיבוד ואחזור של טקסטים עבריים מתמקדת בשני תחומים:

א. ריבוי צורות לערך מילוני

ב. עמימות - ריבוי משמעויות לצורה לשונית

4. פתרונות בשפות אחרות

באנגלית, הבעיה הבסיסית פשוטה יותר. נשווה את הבעיתיות הקיימת בעברית, כפי שתוארה בסעיף 3.1 לעומת השפה האנגלית.

א. המוספיות שבתחילת מילה ובסופה הן קבועות ומספרן מצומצם. (לדוגמא: ING, ED, UN, DIS).

ב. המוספיות אינן כוללות שינוי במבנה הבסיס.

ג. מהיות המוספיות קבועות ומצומצמות לא קיים קושי בזיהוי תחילית הבסיס בשל הוספתן.

ד. היות ובאנגלית אין ניקוד לא קיימת בעיית העמימות בשל כתיב חסר ניקוד.

ה. באופן דומה, לא קיים מצב של ריבוי ייצוגים אורתוגרפיים שונים לאותה מילה (כתיב חסר, מלא ויתיר).

מכיוון שהבעיה פשוטה יותר גם הפתרון פשוט יחסית. הפתרון באנגלית נקרא Stemming והוא מבוסס על מציאת "שורש" המילה על ידי קיצוץ האותיות הסופיות שלה.

לדוגמא, המילים compress, compression, compressed שקולים למילה compress.
[Witten, 1994]

5. Recall לעומת Precision

לצורך הדיון בהבדלים בין הפתרונות, נקדים ונבהיר שני מדדים משמעותיים מאוד בתחום האחזור.

א. *Recall*

מדד זה מבטא את מידת ההתאמה של יחידות המידע הרלוונטיות שאוחזרו מתוך כלל יחידות המידע הרלוונטיות הקיימות.

ובצורה פורמלית:

$$\textbf{\textit{Recall}} = \frac{\textit{items retrieved and relevant}}{\textit{Total relevant in collection}}$$

[Dewire, 1994]

ב. *Precision*

מדד זה מבטא את מידת ההתאמה של יחידות המידע הרלוונטיות שאוחזרו מתוך כלל אוכלוסיית יחידות המידע שנשלפו.

ובצורה פורמלית:

$$\textbf{\textit{Precision}} = \frac{\textit{items retrieved and relevant}}{\textit{Total retrieved}}$$

[Dewire, 1994]

במילים אחרות, ה-Recall מבטא את דיוק השליפה מכלל האוכלוסיה הרלוונטית. וה-Precision מבטא את דיוק התוצאות עצמן.

לא אחת קורה שה-Recall וה-Precision באים אחד על חשבון השני.

6. בעיית העמימות

בעיית העמימות הלקסיקלית נובעת מכך שלצורות זהות בעברית קיימות מספר משמעויות. נוכל להבדיל בין עמימות "קלה" לעמימות משמעותית יותר.

א. צורות שונות של ערך מילוני השונים בכינויי הגוף. כדוגמת ההבדל בין "רוצֵה" ל-"רוצָה".

ב. ערכים מילוניים שונים היוצאים מאותו שורש והנכתבים באופן זהה. לדוגמא: "כְּבוּד" לעומת "כָּבוּד".

ג. ערכים מילוניים שונים הנכתבים באופן זהה. כמו "עֶגְלָה", "עֻגְלָה" ו-"עֲגָלָה".

ד. צורות שונות של ערכים מילוניים: שַׁבְּלוּל שֶׁבַּלּוּל.

7. סוגי הפתרונות

בעבודה נציג שישה פתרונות המבוססים על שיטות שונות. מבין השישה נפרט שלושה פתרונות מרכזיים ונבצע השוואה ביניהם. השיטות מוצגות על פי בסיס מאמרים המתארים את הרקע האקדמי שלהם. יודגש כי שלושת הפתרונות בהם נתמקד יושמו באופן מעשי במנועי אחזור מסחריים.

הפתרונות הינם:

- **שימוש בחוקה**

פתרון המבוסס על יוריסטיקה. הפתרון הוא פרי עבודתו של גדי פנקס, שיושמה באופן מעשי במוצר "קונטקסט". השיטה המוצעת היא הגדרת הפיתוח הלשוני באמצעות אוסף חוקים המופעלים בשלב הניתוח של הקלט ובניתוח השאילתא.

- **שימוש במילון**

שימוש בבסיס נתונים (מילון של מט"ח) ככלי לאנליזה של טקסטים עבריים. תאור הפתרון מתבסס על מאמרם של פרופ' שווקה ואחרים. פתרון זה יושם בפרוייקט השו"ת של אוניברסיטת בר-אילן ובמספר מנועי אחזור מסחריים (XRS, RetrievalWare, Insight into Information)

- **שילוב יוריסטי ומילוני**

שילוב של חוקה ומאגר מידע מילוני-סטטיסטי. הפתרון מוצג במאמר שנכתב ע"י חוקרי המרכז המדעי של יב"מ בחיפה ונארז כמוצר מסחרי בשם HMD. פתרון זה יושם בכלים מסחריים ובהם BabaGuru.

- **פתרונות נוספים**

בנוסף לפתרונות הקודמים יתוארו שלושה פתרונות אפשריים נוספים. הראשון שבהם מוצע ע"י דגן ואיתי, והוא מבוסס על שילוב של מנתח לקסיקלי הכיל מנתח מורפולוגי ומאגר סטטיסטי של צרופים. השיטה מתבססת על ניתוח לקסיקלי של הטקסט וכאשר מתעוררת בעיה של רב-משמעות לקסיקלית פותרים אותה באופן סטטיסטי.

הבדיקה הסטטיסטית נעשית על ידי השוואת הצרופים הלשוניים מול מאגר מידע **מאותו** תחום המכיל מידע סטטיסטי של המצאות צרופים מסוג זה בקורפוס דומה.

השיטה השניה, שונה באופן מהותי מכל היתר, היא שיטתו של פרופ׳ ארנן. השיטה מתבססת על כתיבת עברית בתעתיק לועזי.

השלישית מוצעת ע״י הרץ ורמון והיא מבוססת על אוטומט הקשר-קצר לפתרון בעיית העמימות הלקסיקלית.

8. שימוש בחוקה

השיטה הראשונה למימוש מורפולוגיה עברית היא יישומה בעזרת חוקה. ג׳ פנקס מציג מימוש של מערכת אחזור טקסטואלית המבוססת חוקה. [פנקס, 1985]

הפתרון מתבסס על הרחבה של הצורות המופיעות בטקסט בשלשה מימדים:

א. המימד המורפולוגי

ב. המימד האסוציאטיבי

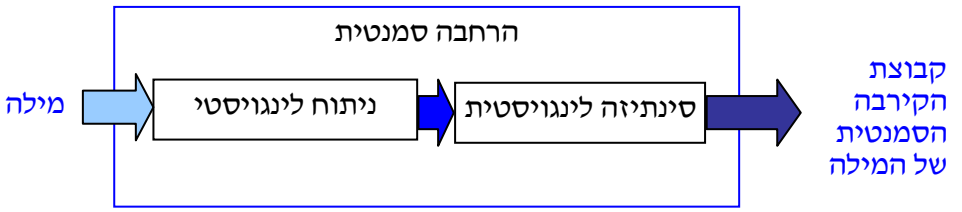
ג. המימד הפונטי

המימד השני, האסוציטיבי, מטפל בקשרים בין מילים. קשרים אלו מבוטאים בעזרת תזאורוס. המימד השלישי, הפונטי, מטפל במילים הנשמעות באופן דומה גם אם הן נכתבות בצורה שונה.

נעיר כי הדיון במימד האסוציטיבי והפונטי אינן במוקד עבודה זו. ולפיכך, נתמקד בצורת הפתרון המוצעת במסגרת המימד המורפולוגי.

המודל מאפשר להגדיר רמות פיתוח שונות עבור כל אחד מהמימדים. ברור כי שככל שתוגדל עוצמת הפיתוח בכל אחד מהמימדים יגדל ה-Recall אולם תקטן רמת ה-Precision.

המודל מתבסס על הגדרת קירבה סמנטית בין מילה בטקסט לבין קבוצת הקירבה הסמנטית שלה.



הגדרת קבוצת הקירבה הסמנטית:

בהנתן עוצמות פיתוח בכל מימד, קבוצת הקירבה הסמנטית של צורת שפה מסוימת F מכילה את כל צורות הטקסט ורק צורות אלו המתפתחות מ-F על פי עוצמה הנתונה בכל מימד..

במילים אחרות, קבוצת הקירבה הסמנטית מכילה את אוסף הפיתוחים בכל עוצמה בשלושת המימדים של צורה נתונה.

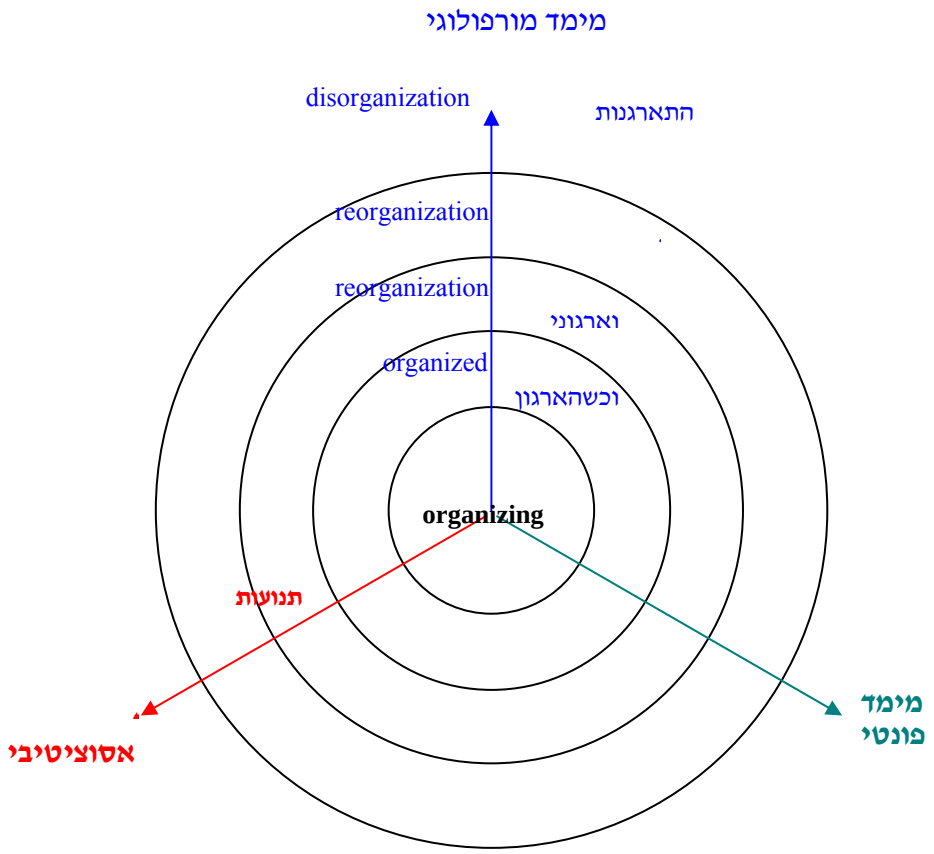
אם נתרכז במימד המורפולוגי, קבוצת הקירבה הסמנטית מכילה את כל הצורות המתקבלות מפיתוח ברמת עוצמה נתונה של צורה נתונה.

פנקס מגדיר חמש רמות של עוצמות פיתוח לשוני עבור המימד המורפולוגי העברי:

א.	המילה בדיוק כמות שהיא	- שומרון	⇦	שומרון
ב.	קידומות	- שומרון	⇦	השומרון, וכשהשומרון
ג.	נטיות השם	- איזור	⇦	איזורים, ואיזורי
ד.	נטיות השם והתואר	- קיבוץ	⇦	קיבוצניקית, הקיבוציות
ה.	נגזרות השורש	- זקן	⇦	הזדקנות, זקנה

באופן דומה, מוגדרות חמש רמות של פיתוח מורפולוגי באנגלית.

תרשים המודל:



מודל הקרבה

מודל הקרבה מתאר באמצעות מודל כללי את בעיית איתור קבוצת הקרבה ופתרונה.

הבעיה הכללית:

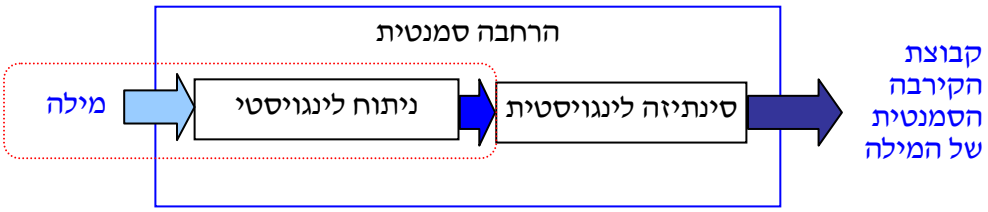
נתון בסיס נתונים טקסטואלי המכיל יחידות מידע המורכבות ממילים.
נתונה שאילתא בוליאנית המכילה מילים ועוצמת קרבה כאשר בין המילים ישנן אופרטורים בוליאניים ומטריים.

המטרה היא איתור מהיר ומדויק של כל המילים בבסיס הנתונים הקרובים מעוצמה i למילים שבשאילתא.

לדוגמא, שאילתא בשפת משתמש: [״ישראל וגם (#עליה או *מדינה)] תתפתח לפעולות:
(Results (״מדינה״,1) Or Results (״עליה״,4)) And Results (״ישראל״,0)

תוצאת השליפה היא אוסף הצבעות על יחידת מידע הכוללות את מילה ״ישראל״ כמות שהיא ושקיימות בהן גם אחד הנגזרות של השורש ״עלה״ או קידומות השם של המילה ״מדינה״.

היות והדיון עוסק בגרעין הניתוח המורפולוגי, העיסוק בסינתזה איינו רלוונטי. באופן דומה, ניתן לומר כי ההיבטים הנוספים של האנליזה (ניתוח אסוציטיבי ופונטי) אינם רלוונטיים אף הם. (החלקים המסומנים הם הרלוונטיים).



פעולות הנירמול

האנליזה והסינתזה מבוצעות באמצעות אוסף פעולות נירמול. פעולת הנירמול מבטאת קשר קירבה ברמה כלשהי בין צורה (מילה מקורית) לבין מילה מנורמלת.

באופן כללי:

$$I\ (item) = \begin{cases} \text{כל הרשימות המנורמלות שניתן להגיע אליהן מ- } \textbf{\textit{item}} \text{ ע״י פעולת נירמול מעוצמה i ומטה.} \end{cases}$$

בניגוד לתפיסה המילונית, שבה פעולת הנירמול הופכת צורה לערך מילוני, בחר פנקס ליישם פעולות נירמול המבוססות על מציאת מחרוזת משותפת גדולה ביותר לכל הצורות שבאותה קבוצת קירבה.

מימוש האנליזה המורפולוגית

לצורך מימוש האנליזה המורפולוגית מוגדרות הפעולות הבאות:

א. הכלה

ב. הפרשים

ג. הכלה מורפולוגית

ד. הכלה מורפולוגית חזקה (בעברית)

הכלה

st1 מוכל ב-***st2*** אם כל תו של ***st1*** נמצא ב-***st2*** באותו סדר אך לאו דווקא ברציפות.
סימון: זק < הזדקנות

הפרשים

ההפרשים ***st2-st1*** הינן כל תת המחרוזות שמוכלות ב- ***st2*** אך אינן ב-***st1***.

הכלה מורפולוגית

st1 מוכל מורפולוגית ב-***st2*** מעוצמה i אם ***st1*** < ***st2*** וההפרשים ***st2*** - ***st1*** הם חוקיים (על פי חוקי המורפולוגיה לרמה ***i***).
סימון: ***st1*** *<i>* ***st2***

הכלה מורפולוגית חזקה בעברית

st1 מוכל מורפולוגית בצורה חזקה ב-***st2***, אם ***st1*** הוא שורש עברי של ***st2***.
סימון: ***st1*** *<4>* ***st2***

האנליזה המורפולוגית

האנליזה המורפולוגית ממומשת באמצעות פעולות נירמול המוגדרות באופן פורמלי בעזרת שפה רגולרית.

החוקים נכתבים בשפה הרגולרית. בעזרת תכנית חילול נוצר אוטומט סופי המבצע את ה-actions המתאימים בהתאם למשקל של מילת הקלט.

9. מימוש במילון

שיטה נוספת למימוש מורפולוגיה עברית היא יישומה בעזרת מילון. פרופ׳ שווקה ואחרים מציגים במאמר משותף את מימוש התפיסה המילונית. הפרק כולו נכתב על בסיס מאמר זה [Choueka, 1978]

המימוש, הקרוי על ידם ״קדמה״, הינו תת פרוייקט במסגרת פרוייקט השו״ת של בר-אילן. קדמה נועד לתת את השכבה המורפולוגית במערכת הכוללת של אחזור מידע בטקסט חופשי.

המחברים טוענים כי המודל הינו כללי ולמרות שהוא תפור ספציפית לשפה העברית, ניתן ליישמו בשפות אחרות ובעיקר בשפות שמיות.

המודל המוצג, התוכנה והמילון שפותחו בעקבותיו מורחבים כיום ע״י מט״ח במסגרת פרוייקט ״מלי״מ״.

עץ המילים The Vocabulary Tree

הגדרות

form- מחרוזת תווים סופית שאיננה כוללת סימני פיסוק או רווחים (״צורה״).

l-form - (language form) ערך כפי שמופיע במילון סטנדרטי או צורה הנובעת מערך זה, כדוגמת הטיות פעלים, צורות ריבוי, כינויי גוף (״מילה חוקית בשפה״ או ״הרכבה״).

t-form - (text form) כל ״צורה״ כפי שמתקבלת מתוך טקסט נתון. צורה מסוג זה עשוייה אמנם להיות ערך מילוני, אך אפשר שהיא שם עצם פרטי, ראשי תיבות, קיצור, טעות דפוס או ערך שאיננו מילוני (סלנג או מילה מקצועית).

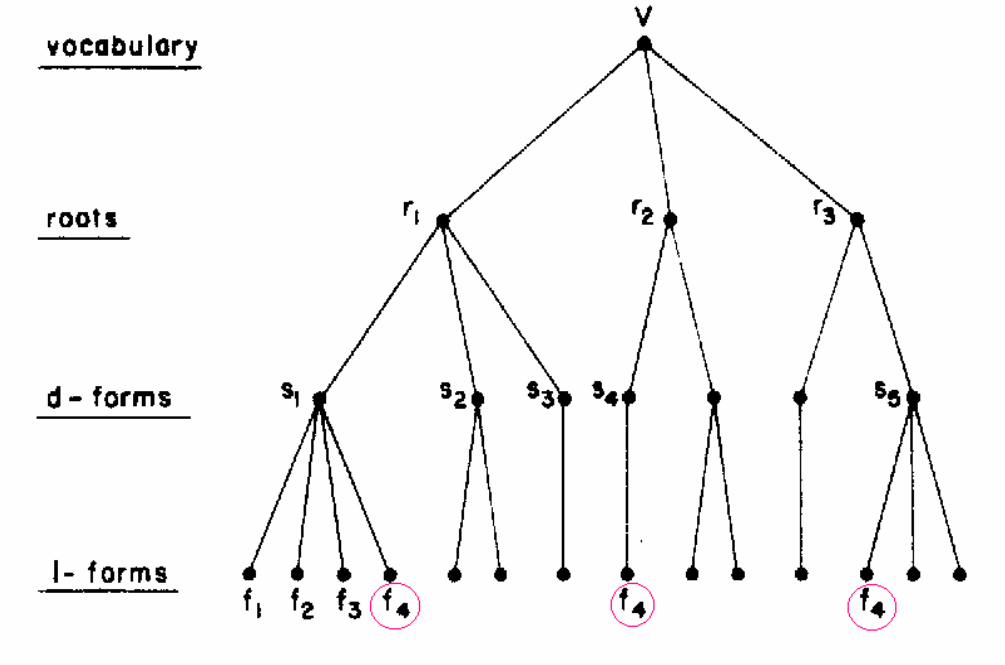
Word - מופע של t-form.

d-form - (standard dictionary form) ערך כפי שמופיע במילון סטנדרטי (״צורה מילונית״ או ״ערך מילוני״). לדוגמא ״זרבובית״ היא הצורה המילונית של ״זרבוביותנו״.

root- (שורש) הצורה המייצגת אוסף מילים אשר קשורים הן מורפולוגית והן סמנטית.

מבנה עץ מילים כללי לשפה

מודל העץ כולל קשרים דו צדדיים בין האיברים השונים. יודגש כי מילה חוקית בשפה (l-form) יכולה להיות קשורה למספר ערכים מילוניים (d-forms). מודגם באיבר f4 באיור.

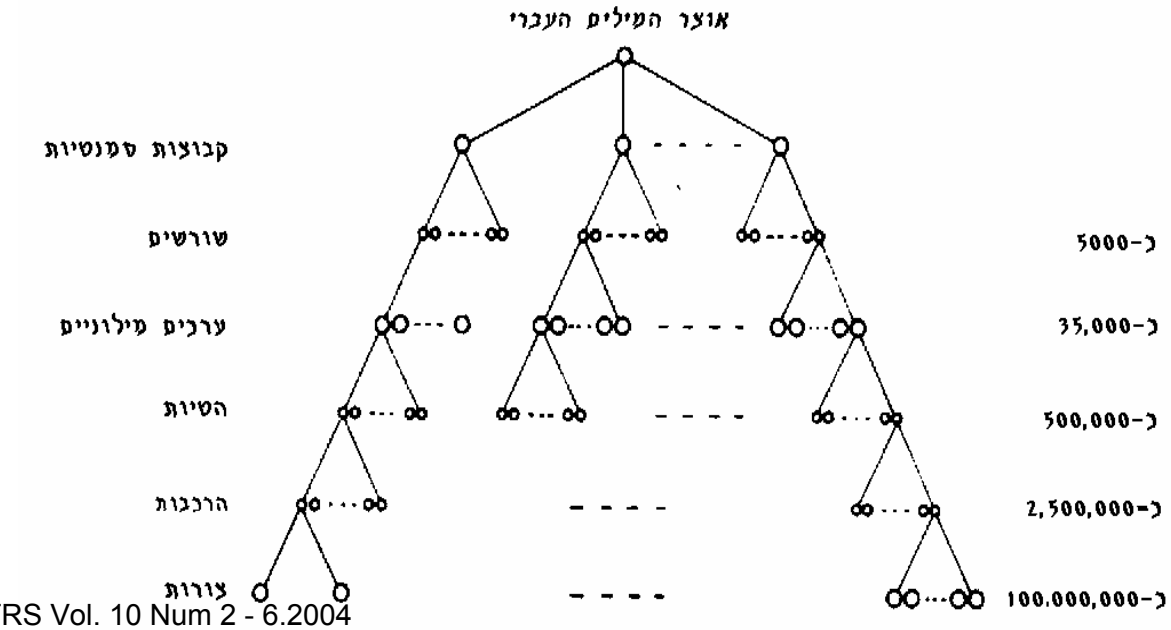


על עץ המילים ניתן לבצע שתי פעולות מורפולוגיות יסודיות:

א. סינתזה - דהיינו הרחבה, מציאת אוסף כל הפיתוחים החוקיים (l-forms של מילה מילונית חוקית (d-form). (s-> F(s)).

ב. אנליזה - הפעולה ההפוכה, מציאת אוסף כל מילות המילון החוקיות של צורה נתונה.

עץ המילים בשפה העברית



מאפייני ההטיה בעברית המהווים שיקול בתכנון פתרון עץ המילים:

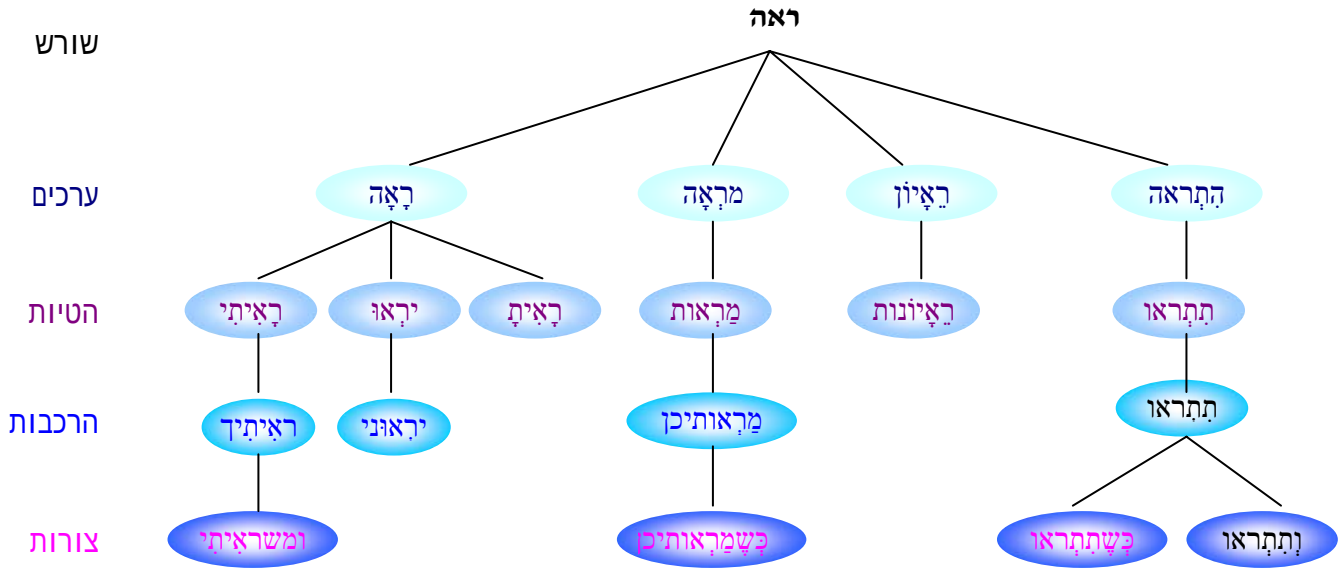
א. מספר קטן יחסית של מילים מילוניות שקשורים עליהם מספר רב של צורות חוקיות. במבנה העץ, החלק התחתון הוא רחב מאוד.

ב. צורות שונות הקרובות לשונית פזורות בכל תחום האלף-בית בשל הקידומות והתוכיות (prefix and infix).

ג. צורות קרובות עשויות להכיל מספר מועט מאוד של אותיות משותפות (לעיתים, רק אות בודדת כמו "בת" לעומת "ולכשבנותינו").

ד. בהשמטת התנועות (vowels) יתקבלו מספר רב של מילים בעלות כתיב זהה אך בעלות משמעות שונה (הומוגרפים).

ניתן להמחיש את משמעות עץ המילים על ידי התמקדות בשורש בודד "ראה"



הפתרון המעשי של מודל העץ מבוסס על היסודות הבאים:

א. רמת ההרכבות במילון העברי היא בגודל סביר יחסית הניתן לניהול (2.5 מיליון צמתים).

ב. ניתן לזהות את ההרכבות הבסיסיות של צורה נתונה על ידי ניתוח הקידומות שלה תוך הסתייעות בחוקים דקדוקיים.

בפועל המודל מגדיר מילון של כל ההרכבות החוקיות בעברית ומנתח את הצורות ע"י הפרדת הקידומות האפשריות:

א. בדיקת חוקיות הקידומת

ב. בדיקת הערך המתקבל מול המילון

ג. בדיקת החוקיות של שילוב ערך המילון עם הקידומת

לדוגמא הצורה "וכשמחשבכם":

	קידומת	ערך	קיום במילון	שילוב קידומת עם הערך	מסקנה
1		וכשמחשבכם	לא קיים	לא רלוונטי	לא קיים
2	ו	כשמחשבכם	לא קיים	לא רלוונטי	לא קיים
3	וכ	שמחשבכם	לא קיים	לא רלוונטי	לא קיים
4	וכש	מחשבכם	קיים	חוקי	קיים
5	וכשמ	חשבכם (החשב שלכם)	קיים	חוקי	קיים
5	וכשמ	חשבכם (חשב עליכם)	קיים	לא חוקי	לא קיים

קבצים לינגויסטים

המילון

הקובץ הבסיסי הוא המילון. קובץ זה מבוסס על מילון עברי-עברי קיים, "המילון החדש" של אבן-שושן. הקובץ מכיל רשומה לכל מילה במילון כשאוסף בלשנים העשירו את המידע הקיים ברשומה בפרטים הבאים:

א. תאור סמנטי קצר;
ב. השורש;
ג. חלק דיבר;
ד. מין;
ה. מספר;
ו. מקור היסטורי ולינגויסטי של המילה (מקראי, תלמודי, ימי-ביניימי, מודרני);
ז. קידומות אפשריות;
ח. סבירות לתוספות של כינוי-השם;
ט. קוד ליצירת צורת נקבה, זוגית וריבוי;
י. בניין;
יא. צורות איות נוספות.

קובץ תרגום ערך מילוני לצורה מורכבת - d-forms to Compounds File (DC)

קובץ ההרכבות המשמש לתרגום ערך מילוני לצורה מורכבת. הקובץ נבנה ע"י עיבוד קובץ המילון. הקובץ מכיל את פרטי המידע הבאים:

א. צורות הכתיב השונות של הערך;
ב. הטיות (על פי הנטיות);
ג. הרכבות (על פי נטיות הפועל);
ד. צורת כתיב שונות של המילים המורכבות.

לדוגמא המידע שהקובץ יכיל עבור הערך "רָאָה":
לערך צורת כתיב יחידה.
ההטיות "ראיתי", "יראו", ... "ראיתם".
על פי ההטיות את ההרכבות "יראוני", "ראיתכם", "ראתיכן" ועוד.

קובץ תרגום מילה מורכבת לערך מילוני - Compounds to d-form File (CD)

קובץ מהופך ל-DC המשמש לתרגום ערך מילוני לצורה מורכבת.

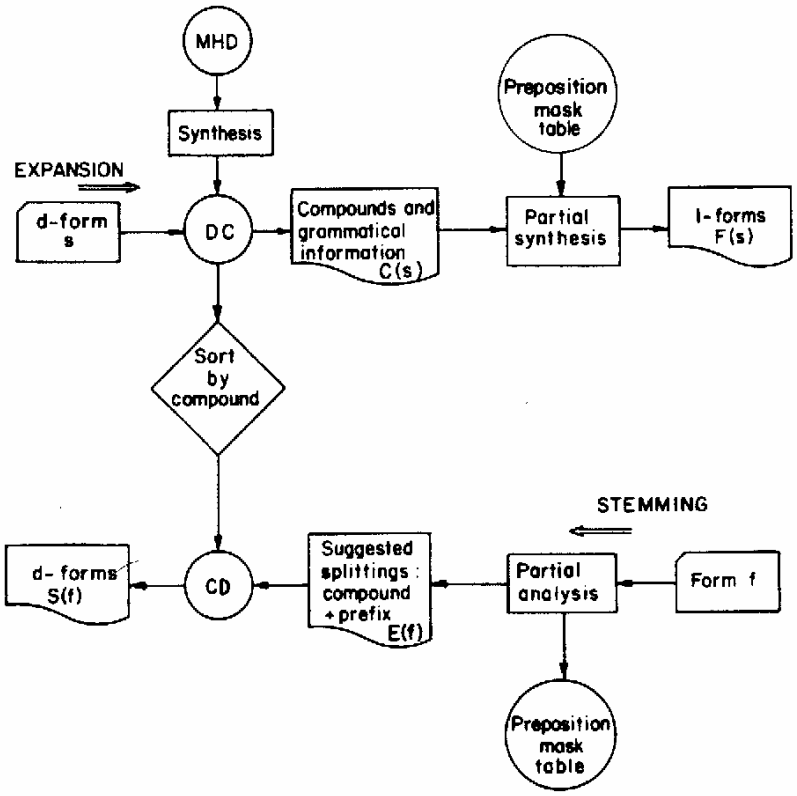
המידע הקיים בקובץ:

א. המילה המורכבת עצמה;
ב. הצבעה לערכי ה-"אבות" המילוניים (הערכים המילוניים מזוהים ע"י מספר);
ג. מידע דקדוקי נוסף.

עבור המילה המורכבת "וכשמראיכן" תופיע בקובץ המילה עצמה, הצבעות לערכים המילוניים "מַרְאֶה", "מַרְאָה" (ראי), "מַרְאָה" (חזיון).

לצרכים מעשיים, ניתן ליצור קבצים חלקיים המכילים רק את הערכים הנובעים מהמילים המופיעות בטקסט.

תהליך האנליזה



האיור מבטא את תהליך בניית הקבצים העיקריים ואת תהליכי האנליזה והסינטזה. קובץ ה-DC נבנה על ידי עיבוד המילון (MHD). ה-CD הוא הקובץ המהופך והוא נבנה על ידי מיון הקובץ על פי ההרכבות.

התהליך העליון, הסינטזה, מתבצע בזמן אחזור. עבור מילה במילון (לדוגמא "ראיון") יפותחו בעזרת קובץ ה-DC אוסף כל ההרכבות שלה (ראיונותי, ראיונותיהם...). קבלת אוסף כל הצורות נעשית על ידי הוספת הקידומות למכל הרכבה על פי המידע הלשוני הקיים עבור כל הרכבה (וכשהראיון, כשראיונותיכם אבל לא "הראיונתיכם").

התהליך ההפוך, האנליזה, מוצג בחלק התחתון של האיור. תהליך זה בא לידי ביטוי בניתוח הקלט לאינדוקס. עבור כל מילה שמתקבלת בטקסט (לדוגמא: "וכשהפיל") מורדות הקידומות שלה והערך נבדק מול קובץ ה-CD על מנת שניתן לזהות הרכבה חוקית (במקרה שלנו "הפיל" ו-"פיל") התוצאה הסופית היא הצבעה על הערכים המילוניים ("נפל" ו-"פיל").
היות והקובץ מכיל רק את ההרכבות החוקיות יזוהה הערך "נפל" בלבד עבור "הפילם" והערך "פיל" עבור "הפילים".

פתרון בעיית העמימות

הניתוחים המורפולוגיים המתקבלים כתוצאה מניתוח מילה בודדת הינם מדויקים. אולם בעיית העמימות נותרת בעינה, היות ולאחר האנליזה אפשר שיתברר שלערך מנותח יש יותר מניתוח חוקי יחיד.

בעיית העמימות נפתרת על ידי הרחבת המודל. הניתוח המורפולוגי נותר כפי שהוא ואילו ההחלטה על סבירות הניתוח נקבעת על פי ההקשר.

הרעיון הוא שאוסף חוקים מבטא את חוקיות הקשרים בין המילים באותו משפט, כך שהתוצאה המנותחת מושפעת מסביבת המילה המנותחת.

אם נחזור לדוגמא שהבאנו בסוף הסעיף הקודם, את הצורה "וכשהפיל" ניתן לזהות בשני ערכים מילוניים ("נפל", "פיל"). סביבת המילה המנותחת יכולה להבהיר את המשמעות המדויקת. הפועל "עמד" בצרוף "**וכשהפיל** עמד" יכול ללמד שהכוונה לשם עצם ומכאן שהניתוח "פיל" הוא רלוונטי ואילו הניתוח על פי הפועל "נפל" איננו רלוונטי.

הבאנו כדוגמא צורה שלה שתי משמעויות מילוניות אולם ריבוי משמעויות היא תופעה רווחת מאוד בשפה העברית וישנן צורות שלהן יותר מעשר משמעויות מילוניות שונות.

10. שילוב יוריסטי ומילוני

השיטה השלישית היא שילוב של ניתוח מורפולוגי על בסיס חוקה עם בסיס מידע סטטיסטי להפגת העמימות הלקסיקלית.

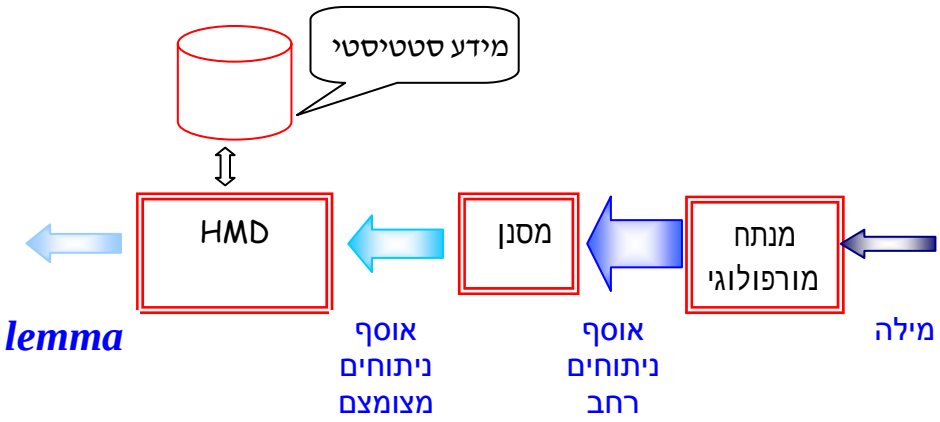
השיטה מתוארת במאמרם של ד"ר כרמל וד"ר מארק ממעבדת המחקר של יב"מ בחיפה [Carmel, 1999].
בבסיס השיטה עומדת העובדה כי ל55%- מהמילים העבריות יש יותר ממשמעות אחת.

מבנה הפתרון

הפתרון מורכב משילוב של שלושה רכיבים:

א. מנתח מורפולוגי
ב. מסנן
ג. "מפיג עמימות"

תאור גרפי של מבנה הפתרון:



המנתח המורפולוגי

השלב הראשון בתהליך הוא הניתוח המורפולוגי. שלב זה מבצע ניתוח על פי חוקה, באופן דומה לשיטה הראשונה שהוצגה בעבודה זו.

הקלט לתהליך הוא מילה בעברית והפלט הוא אוסף ניתוחים מורפולוגיים. באופן מעשי, מתבצע שלב זה באמצעות מוצר קיים הקרוי "אבגד".

המסנן

שלב הסינון מקבל את אוסף הניתוחים המורפולוגיים שהתקבלו מהשלב הקודם, ופוסל את אותם התוצאות שאינו סבירות מבחינת החוקים המורפולוגיים.

התוצר המתקבל משלב זה הוא אוסף מצומצם של ניתוחים מורפולוגיים.

מפיג העמימות

השלב השלישי, HMD (*Hebrew Morphological Disambiguator*), מבצע הערכה הסתברותית לכל ניתוח על ידי הסתייעות במאגר מידע סטטיסטי. המודל מאפשר

להגדיר פרמטר f המבטא את היחס precision/recall. רק ניתוחים שציונם גבוה מהנדרש יועברו כפיתוחים סבירים מבחינת חד-משמעיות.

התוצר של תהליך זה הוא אוסף של lemma (הצורה הבסיסית) אחת או יותר שסבירות זיהוי המשמעות שלהן היתה גבוהה.

האלגוריתם לפתרון העמימות

```
Context-free-HMD( word w, threshold ε)
  Analysis[] ← Avgad(w)
  n ← |Analysis|
  if n = 0 /* for illegal words we use the input word as a base form.*/
    return (w,1)
  else
    if n = 1 /* one analysis: no dilemma, use its lemma as a base form */
      return (Analysis[1].lemma,1)
    else /* more than one analysis */
      Lemmas ← {}
      for i = 1 to n
        let Freq[i] be the relative frequency of pattern(Analysis[i])
        if Freq[i] ≥ ε
          Lemmas ← Lemmas + (Analysis[i].lemma, Freq[i])
      return Lemmas
```

נדגים את האלגוריתם בשלושה מקרים:

א. מילה שאיננה חוקית
כשהמנתח הדקדוקי מזהה כמילה שאיננה "חוקית" הניתוח המוחזר הוא המילה המקורית עצמה. יודגש כי הכוונה ב"חוקיות" המילה היא ליכולתו של המנתח הדקדוקי לזהותו ולא לחוקיותו בשפה העברית. לדוגמא השם הפרטי "אלכסנדר" לא יזוהה כמילה חוקית.

ב. ניתוח יחיד
כשתוצאת הניתוח הדקדוקי היא ערך יחיד לא קיימת כלל בעיה של עמימות. לדוגמא המילה "המנעול" שלה משמעות יחידה, תוחזר כ- lemma "מנעול".

ג. ניתוח רב משמעי
למילה "משטרה" שלושה ניתוחים שונים: "משטרה", "משטר" ו-"שטר". מאגר המידע הסטטיסטי יכול ללמדנו שסדר ההסתברויות של הניתוחים הוא: "משטרה", "משטר" ,"שטר". על פי פרמטר היחס f המבטא את היחס precision/recall, יכול המודול של הפגת העמימות לקבוע את הניתוחים שעברו את הסף f כניתוחים סבירים. אם יועבר סף גבוה תקבע המילה "משטרה" כניתוח יחיד. בסף נמוך יתווספו המילים "משטר" ו-"שטר" כסבירות.

הקמת המאגר הסטטיסטי

המאגר הסטטיסטי נבנה על ידי מעבר ידני על 16,000 ערכים. תוצאת הניתוח הממוחשב קושרה עם הניתוח "האמיתי" של הערך ונרשמה במאגר נתונים.

11. פתרונות נוספים

בנוסף לפתרונות שהוצגו בסעיפים הקודמים קיימים פתרונות נוספים אותם ננסה להציג בקצרה.

11.1 ניתוח על בסיס סטטיסטי

דגן ואיתי מציגים גישה שונה המבוססת על ניתוח סטטיסטי [דגן, 1992]. הרעיון העומד מאחורי תפיסה זו הוא פתרון בעיית רב המשמעות באמצעות ניתוח סטטיסטי של קורפוס טקסטואלי העוסק בנושא ספציפי. השיטה עצמה איננה תלויה בשפה העברית וניתן לממשה בכל שפה.

למימוש הפתרון יש צורך בשני מרכיבים: מנתח תחבירי ולקסיקון.

המנתח התחבירי

המנתח התחבירי מקבל כקלט משפט ומוציא כפלט עץ ניתוח תחבירי. קיימת אפשרות שיתקבל כפלט יותר מעץ ניתוח אחד וזאת במקרים שהמשפט הוא רב משמעי. המנתח התחבירי כולל בתוכו גם מודול מורפולוגי. נעיר כי למימוש השיטה ניתן להשתמש במודול מורפולוגי המיושם על פי אחת השיטות שתוארו בפרקים הקודמים. כך ששיטה זו איננה בהכרח חליפית לאחרות אלא יכולה להוות סינתזה עם חלקן.

כל עלה בעץ הניתוח התחבירי מכיל את צורת הבסיס של המילה כפי שהתקבל מהמנתח המורפולוגי בצרוף חלק הדיבר שלה.

הלקסיקון

מסד הנתונים נבנה עבור קורפוס טקסטואלי העוסקים באותו תחום. מסד הנתונים נבנה בשיטה הבאה: ראשית מריצים את המנתח התחבירי על טקסטים רבים מהתחום הרצוי. תוצאת ההרצה היא אוסף גדול של עצי ניתוח. בכל עץ שבאוסף מאתרים את צרופי המילים המופיעים בו ביחסים תחביריים (נושא-פועל, פועל-מושא ....). הצרופים שזוהו נשמרים במסד הנתונים.

עבור כל צרוף נבדק אם הוא מופיע בכל עצי הניתוח של המשפט. אם כן המשמעות היא שהצרוף תקין מבחינה סמנטית. אם הצרוף מופיע רק בחלק מהניתוחים המשמעות היא שהצרוף תקין בספק. במסד הנתונים נשמר מידע סטטיסטי לשני סוגי המופעים.

פתרון בעית רבוי המשמעויות

פתרון בעיית רב המשמעויות תלויה בתוצאות הניתוח הסטטיסטי. המודל מבוסס על ההנחה כי לצרוף תקין מבחינה סטטיסטית יצוין במסד הנתונים במספר גבוה של מופעים שזוהו כ-"תקינים" ו-"תקינים בספק". בעוד שלצרוף שאיננו תקין יהיו מספר מופעים קטן של מופעים המוגדרים כ-"תקינים בספק".

המחברים מביאים כדוגמא את המשפט "לקחתי את הלחם משולחן ואכלתי אותו" כאשר קיימת עמימות לגבי הצרוף נושא-פועל. סביר שבמסד הנתונים יופיעו מופעים רבים של "לאכול לחם" לעומת הצרוף "לאכול שולחן" שאיננו קיים.

כך שלטענת המחברים אלגוריתם המבוסס על שיקול סטטיסטי יביא בסבירות גבוהה את האפשרות הניתוח הנכונה.

לסיכום, נעיר כי השיטה המוצעת מבוססת על ניתוח טקסט מתחום אחד ואין זה ברור איך היא מתמודדת עם הבעיה הכללית של ניתוח טקסטים ללא תלות בחומר המקור. אולם, שיטה זו מטפלת בתחומים ששיטות אחרות מתקשות לטפל. אם נקח את הדוגמא שהביאו המחברים, "לקחתי את הלחם מהשולחן ואכלתי אותו". הרי שלמילים "אכלתי", "לקחתי", "מהשולחן" ו-"אותו" אין בעיית עמימות (למעט המילה "הלחם"). העמימות של המשפט היא ברמה הלקסיקלית. לדוגמא, במימוש אחזור משפט זה על פי המודל של דגן ואיתי, בדיקת מי מהשאילתות "אכלתי לחם" או "אכלתי שולחן" יאתרו אותו.

11.2 כתיבה בתעתיק לועזי

פרופ' ארנן [Ornan, 1997] מציע שיטה לכתיבה עברית באותיות לועזיות. היתרון ששיטה זו מרחיבה את מרחב התווים המייצגים כתיב ללא ניקוד.

טבלת התרגום

א	`
ב	B
בּ	B
ג	G
ג'	G'
ד	D
ה	H
ו	W,o,u
ז	Z
ז'	Z'
ח	X
ט	@
י	I,y
כ,ך	K
כּ,ךּ	K

ל	L
מ,ם	M
נ,ן	N
ס	S
ע	&
פ,ף	P
פּ	P
צ,ץ	C
צ'	C'
ק	Q
ר	R
שׁ	$
שׂ	$'
ת	T

מעיון בשיטה עולה כי לכל אות באלף-בית העברי קיימת מקבילה המיוצגת ע"י אות אנגלית. אין ייצוג לאותיות הסופיות (כמנפ"ץ) ולא קיימת הבחנה בין אותיות דגושות לשאינן דגושות ("**ב**" לעומת "**בּ**").

לדוגמא, מקור המשפט "BRI DBR &BRIT&" הוא "עברי דבר עברית".

המקרים היחידים בהם קיימת יותר ממקבילה אחת הם ו"ו, יו"ד ושי"ן. שלהם מקבילות התלויות בשימוש. את ו"ו מייצגות האותיות w,o,u כאשר o,u משמשות כתנועות (לדוגמא, "**ו**הרוח"="**w**hr**u**x"). באופן דומה משמשות האותיות i,y על פי ההקשר כמקבילות של יו"ד ("יין"="iyn"). בניגוד לכתיב חסר הניקוד שאיננו מבחין בין שין ימנית לשמאלית, מציע ארנן שתי חלופות (שׁ=$, שׂ='$).

השיטה מבוססת על כך שלאותיות בהם קיימת עמימות גבוהה (ו,י,ש) יש יותר ממקבילה אחת כך שרמת העמימות בהקשר זה נמוכה יותר.

לדוגמא: כָּבָשׁ = KB$ לעומת כָּבָשׁ = 'KB$. אולם, אין הבדל בין כָּבַשׁ (KB$) ו- כָּבָשׁ (KB$).

השיטה זכתה להכרה כתקן מוכר של מכון התקנים האמריקאי (ANSI Z39.25-1975), אולם לא זכתה להצלחה מעשית. שימוש דומה לשיטה זו ניתן למצוא בטקסטים מדעיים. לדוגמא מאמרו של פרופ' שוייקה [Choueka, 1978] שהובא בפרק 9 בעבודה זו, נכתב באנגלית והדוגמאות העבריות שבו נכתבו בתעתיק דומה לזה של ארנן. השימוש בכתיב לועזי נועד לפתור את בעיית הכתיבה בעברית בטקסט אנגלי.

נעיר כי נראה שהשיטה תורמת חלקית בלבד לפתרון בעיית העמימות הנובעת מריבוי משמעויות לצורה. ואיננה פותרת כלל את בעיית ריבוי הצורות לערך מילוני. השימוש בשיטה איננו פרקטי היות והיא מתמודדת בטקסטים הכתובים על פי השיטה. אם נציע "לתרגם" טקסטים בעברית לשיטת ייצוג זו נצטרך להתמודד עם בעית העמימות בשלב התרגום.

נעיר כי בנוסף לכך השיטה מעלה שאלות פילוסופיות-ערכיות החורגות מתחום של השילוב של בלשנות ומחשבים, על מקומו של הכתב העברי כסמל לאומי וכמייצג את התרבות וההיסטוריה היהודית. הבעיה הערכית מזכירה בעיה אחרת מתחום הכלכלה. בה הוצע לפתור את בעיית האינפלציה באמצעות דולריזציה וההצעה נפלה מסיבות ערכיות, על מקומו של השקל כמטבע לאומי.

נוסיף ונעיר כי קיימת שיטת ייצוג באותיות עבריות המתמודדת עם בעיית הכתיב חסר הניקוד. שיטה קיימת כבר מאות שנים וקיימת עד היום בשפת האידיש. בשיטה זו מיצגות אותיות מסוימות את התנועות.

11.3 עברית כשפה פורמלית

פתרון נוסף הוא הגדרת כוללת של השפה העברית כאוסף חוקים פורמליים וניתוח הטקסט באמצעות אוטומט.

הרץ ורמון מציגים במאמרם [הרץ, 1992] שימוש באוטומט לזיהוי הקשר הקצר ככלי לפתרון בעיית העמימות. לטענת המחברים הכלי מסוגל לזהות שפה שהיא קירוב של שפה טבעית. רמת הקרבה לשפה תלויה ברמת הפירוט של האוטומט.

אחד היתרונות בשיטה המוצעת על ידם הוא שהשיטה איננה תלויה בשפה, כך שניתן עקרונית לממשה בכל שפה שהיא ובכל מרחב יצוג של תגים.

בנוסף, ניתן לשלב בשיטה ידע בלשני על מנת לשפרה. השיטה איננה מבוססת על מאגר מידע (מילון) אלא על מבנה פורמלי מצומצם יחסית.

לסיכום, היתרון בשיטה זו שהיא תוקפת את הבעיה בעיקר במישור הסמנטי. אולם תנאי מוקדם לכך הוא ניתוח מורפולוגי של היחידה הבסיסית, המילה.

12. קריטריונים להשוואה

ההחלטה על הקריטריונים שעל פיהן תעשה ההשוואה, התבססה על שילוב של יכולות פונקציונליות הממומשות כיום בכלים מסחריים יחד עם היכולות הפונקציונליות המעשיות והתיאורטיות שתוארו בחלק מהמאמרים עליהם מבוססת העבודה והכרות מעשית עם תחום זה.

12.1 פונקציונליות - דיוק לשוני

א. יחיד ורבים

השוואת התמיכה בזיהוי של צורות היחיד והרבים בפעלים ובשמות עצם. בנוסף, בדיקת ההתייחסות בטיפול בשמות עצם פרטיים. בדיקת המקרים השגרתיים וגם יוצאי דופן כמו בת-בנות, בר-בני. בנוסף לצורת הריבוי בדיקת הזוגיים (כפל-כפליים, קומה-קומתיים, לעומת מספר-מספריים).

ב. זכר ונקבה

בסעיף זה יושווה הזיהוי של צורת הזכר והנקבה בפעלים ובשמות עצם. בנוסף לצורת הנקבה הקלאסית של המוספיות "ה" ו-"ות", הבחנה בזיהוי זהה של "אשה"-"נשים"- "נשות" (בנסמך), זיהוי שונה לצורת "בן"-"בת" ו-"בן"-"בנות" (לעומת "חתול"-"חתולות").

ג. הטיות פעלים רגילים

השוואת הטיפול בהטיה של פעלים משולשים רגילים בכל הבנינים (שפט, אשפוט, ישפוט...) פעלים אלו מבוססים על השורש המקורי אשר לו נוספים תחיליות ומוספיות. בסעיף זה נכלול גם את השוואת איכות הזיהוי של שורש מתוך מילה מורכבת.

ד. הטיות פעלים עם גזרה מיוחדת (החסרים הנחים והכפולים)

ההשוואה בסעיף זה נוספת לזו שבסעיף הקודם בבדיקת הטיפול בהטיות הגורמות לשינוי בצורת השורש הבסיסי. בסעיף זה נבדוק גזרות כמו חסרי פ"י, פ"נ, נחי ל"ה, וכפולים. לדוגמא: שורש י.ש.ב: ישב, לשבת, אשב, שב! (חסרי פ"י). שורש ס.ב.ב: להסב, נסבותי, אסב (כפולים).

ה. פעלים מיוחדים (מרובעים, מחומשים)

צורות נוספות של פעלים יחודיים הן המרובעים והמחומשים. לדוגמא: לטלפן (מרובע) או טלגרפתי (מחומש).

ו. מילים חדשות

בעוד שבדיקות הפעלים בוחנות בעיקר את ההתמודדות של השיטות היוריסטיות, בדיקת הטיפול במילים חדשות בוחנות את הגבולות של שיטת המילון. ההשוואה בסעיף זה תהיה עבור מילים חוקיות בשפה אשר אושרו על ידי האקדמיה (לדוגמא שוהי, נמלול, כילול, שידרוג, תחיבה).

ז. מילים לועזיות

באופן דומה, נדרש להשוות את ההתייחסות למילים לועזיות "תקניות" המופיעות במילון כמו אורינטציה, אינסטלטור או קרנבל. וכן פעלים שנבנו על מלים לועזיות כמו לפקסס.

ח. סלנג

שני הסוגים של הסלנג (עגה), המקצועי והעממי קנו להם אחיזה ומופיעים בטקסטים. לפיכך נדרשת השוואה של השיטות השונות בהתמודדות עם תחום זה. (לדוגמא: לקמפל, פיקשוש).

ט. כתיב מלא וחסר

השוואת ההתמודדות עם ההבדלים בין כתיב מלא וחסר, לרבות ההתמודדות עם כתיב שגוי מבחינה דקדוקית אך רווח מבחינת השימוש בו.

י. אותיות שימוש (מש"ה וכל"ב)

בנוסף לזיהוי אותיות השימוש כחלק מהטיפול בפעלים ושמות עצם. נדרשת השוואת יכולת הזיהוי וההתמודדות עם אותיות שימוש כתחיליות של שמות עצם פרטיים (לדוגמא: וכשל**יוסף**, מ**לאה**).

יא. צורות חוקיות שאינן נפוצות בעברית מודרנית

היות ועברית הינה שפה חייה ונוספים לה מילים חדשות, אך לעיתים פוחת השימוש במילים מסוימות. יתכן ובטקסטים מסויימים נוכל לדעת מראש כי מילים ספרותיות או מילים שצורתן השתנתה אינן נכללות בה.

12.2 פונקציונליות דיוק הפגת העמימות

אחת הבעיות הבסיסיות שהצגנו היא התמודדות עם העמימות. העמימות של ריבוי הצורות לערך לשוני נבדקה בסעיף הקודם. בסעיף זה נתמקד בהגדרת הקריטריונים להשוואת ההתמודדות עם ריבוי המשמעויות לערך בודד.

לצורך כך נשווה את השיטות על בסיס ארבעה קריטריונים:

א. השוואת מידת הדיוק של ניתוח הצורות הנכונות בלבד (לדוגמא הצורה "מספרי" הינה הטיה של "סֵפֶר", "סַפָּ·ר", "סַפָּר", "מִספר", "מִסְפָּרַיִם" ועוד אך לא של "סיפור", "מְספרה").

ב. השוואת הפגת העמימות באופן לקסיקלי. סינון הניתוח הנכון מקרב הניתוחים האפשריים על פי חוקי השפה. לדוגמא המילה "מספרי" במשפט "נא **מספרי** את הדפים" משמעותו "מַסְפֵּרִי" (קבעי מספר לכל דף). כי לפני מילת היחס "את" צריך להיות פועל.

ג. היות ואפשר שיהיה למשפט יותר מניתוח לקסיקלי אחד תקין. יש צורך בהשוואה האם השיטה מפיגה עמימות על פי הסבירות ההגיונית של המשפט. לדוגמא ניתוח המילה "**מספרי**" שבמשפט "אחד מספרי שבמדף", עשוי לזהות

כתקין מבחינה תחבירית הן את הניתוח "סֵפֶר" והן את הניתוח "סַפָּ·ר". כאשר "הגיונית" ברור שהכוונה היא ל-"סֵפֶר".

ד. הרמה העליונה של ההשוואה היא ניתוח משמעות של משפט מורכב. אם נחזור לדוגמא שהציגו הרץ ורימון, ("לקחתי את הלחם מהשולחן ואכלתי אותו"). ההשוואה הנדרשת היא האם יאוחזר המשפט בשאילתה מסוג "לאכול לחם" (כן) ו-"לקחת שולחן" (לא).

12.3 מימוש

בסעיף זה נגדיר מספר קריטריונים להשוואת הישימות של הפתרונות.

א. קלות מימוש

השוואת מידת הקושי של מימוש הפתרון. בעיקר מבחינת סדרי הגודל של היקף הפיתוח הנדרש לכל פתרון.

ב. יכולת ההרחבה לשפות נוספות

הפתרונות הוצגו במסמך זה בדגש על השפה העברית. היות והן הצורך בטיפול לשוני והן בעיית העמימות קיימים בשפות נוספות, ננסה להשוות את היכולת להרחיב את המודלים השונים לשפות נוספות (שמיות ואחרות).

ג. תשתית למימושים נוספים

במסמך זה התמקדנו בעיקר על מימושים של ישומי אחזור. ננסה להשוות את הרחבת השימוש לישומים נוספים בתחום (לדוגמא: ניקוד טקסטים, הקראת טקסטים, בדיקת איות, בדיקת דקדוק).

12.4 אפקטיביות בתחום האחזור

בפרק 6 במסמך צוינו שני מדדים משמעותיים לבדיקת דיוק האחזור. בסעיף זה ננסה להשוות מימוש המודלים בתחום האחזור על ידי בחינת מדדי הדיוק. יודגש כי ההשוואה בסעיף זה היא בעיקרה תיאורטית ועשויה להשתנות על פי היישום הממשי.

א. *RECALL*

השוואה תיאורטית של מידת ההתאמה של יחידות המידע הרלוונטיות שיאוחזרו בשיטות השונות מתוך כלל יחידות המידע הרלוונטיות הקיימות.

ב. *PRECISION*

השוואה של מידת ההתאמה של יחידות המידע הרלוונטיות שאוחזרו מתוך כלל אוכלוסיית יחידות המידע שנשלפו.

13. השוואה

בפרק זה יושוו שלושת הפתרונות העיקריים על פי הקריטריונים שפורטו בפרק הקודם. באשר לשיטת ההשוואה, חלקה נעשה באופן מעשי וחלקה באופן תיאורטי.

הבדיקות הפונקציונלית של הדיוק הלשוני נעשו בכל שלוש השיטות באופן מעשי:

א. שיטת החוקה נבדקה ע"י ביצוע אוסף שאילתות של "קונטקסט", המוצר בו היא פותחה. השאילתות התבססו על הדוגמאות שתוארו בסעיף 12.1.

ב. עבור שיטת המילון נבדקו שני מוצרים שונים לאחזור מידע טקסטואלי המממשים באופנים שונים את הטיפול המורפולוגי באמצעות ניתוח שנעשה ע"י המוצר "רב-מילים" של מט"ח.

ג. באשר לשיטת השילוב, היות ולצערנו לא היה בידנו פתרון מסחרי כולל של HMD, הסתפקנו בבדיקה מעשית של גרעין הניתוח המורפולוגי שעליו מתבסס הפתרון. גרעין זה נקרא כאמור "אבגד" והוא משולב בתוכנת האחזור ***Insight into Information***. הבדיקה המעשית נעשתה בתקליטור "עבודה ועוד" המבוסס על המוצר.

ההשוואה של פונקציונליות דיוק הפגת העמימות נעשתה על ידי השוואה תיאורטית של שלושת השיטות. נעיר כי לגבי שיטת החוקה לא הוצע כלל פתרון לבעית העמימות הלקסיקלית.

השוואת המימוש נעשתה על פי המאמרים המתארים את השיטות. סעיף ההשוואה האחרון, האפקטיביות בתחום האחזור, נעשה על בסיס ניתוח תיאורטי של השיטות.

13.1 פונקציונליות - דיוק לשוני

תחום	חוקה	מילון	שילוב
יחיד ורבים	המקרים "הקלאסיים" מטופלים בקלות. אם יתבצע טיפול במקרים יוצאי דופן הוא יהיה גורף ויפגע ב-PRECISION. מהבדיקה עולה כי אין הבדל מבחינת החוקה בין "כפל-כפליים" ל-"מספר-מספריים". לא זוהו הריבויים: בת-בנות, עיר-ערים, אשה-נשים, בר-בני, איש-אנשים. לעומת זאת זוהו בן-בנות.	היות וכל המידע קיים במילון הטיפול הוא מלא. לרבות הטיפול בזוגיים וביוצאי דופן.	הניתוח הבסיסי טוב יותר מהחוקה. זוהה ההבדל בין כפליים לכפלים. וכן זוהו הריבויים: בת-בנות, איש-אנשים. למרות זאת לא זוהה אשה-נשים.
זכר ונקבה	הטיפול בפעלים הוא טבעי וניתן לממשו באמצעות החוקה. אין הבדל מבחינת החוקה בין הקרבה "בן-בנות" לעומת "חתול-חתולות".	מטופל באופן מלא על בסיס המידע הקיים במילון.	באופן כללי דומה לשיטת החוקה. אולם התוצאה המעשית מעט טובה יותר.

תחום	חוקה	מילון	שילוב
הטיית פעלים רגילים ומיוחדים	החוקה מבטאת את ההטיות בכל הבנינים גם כאלו שאינן חוקיות דקדוקית. בבדיקה המעשית התברר כי נתמכים גם הגזרות המיוחדות כמו פ״י וכפולים ישב-אשב-לשבת-שב, סבב-להסב-נסבותי - אסב. וכן נתמכים השורשים שאינם משולשים כמו אטלפן וטלגרפתי.	מלא. רק הערכים הרלוונטים. לדוגמא לשכן-משוכן לעומת למשכן. נתמכים גם הפעלים בעלי הגזרות המיוחדות וכן פעלים מרובעים ומחומשים (הקיימים במילון)	נתמך באופן מלא בדומה לחוקה. המקרה היחידי שלא זוהה הוא שב בפועל י.ש.ב.
מילים חוקיות חדשות וסלנג	היות ולא קיים מילון. אין הבדל בין מילים חדשות לקיימות.	מילים חדשות נוספות למילון. כולל מילות סלנג. אולם מילה שלא הוגדרה במילון איננה מטופלת דקדוקית בשום צורה. מבחינה לשונית היא זהה לשם עצם פרטי. במילון קיימים הפעלים לפקסס ולפקשש	היות והמילון הוא חלקי. מילים חדשות זהות למילים שאינן קיימות במילון ומטופלות לפיכך בהתאם.
מילים לועזיות	מילים לועזיות יטופלו באופן דומה למילים עבריות.	מילים ממקור לועזי הקיימות במילון מטופלות באופן מלא. מילה לועזית שאיננה קיימת במילון הרי היא כמילה חדשה.	מילים לועזיות נפוצות עשויות להיות קיימות במילון הסטטיסטי. היתר יטופלו על פי החוקה.
כתיב מלא וחסר	מטופל במסגרת החוקים הכלליים. בחלק מהמקרים הטיפול עלול להיות שגוי.	הקשרים בין מילה בכתיב מלא וחסר מבוטא במילון. המילון ״סלחני״ לטעויות ״קלות״ הרווחות בציבור.	מטופל במסגרת החוקים. אך כאמור משופר במסגרת המילון ומנגנוני הסינון.
אותיות שימוש	כנ״ל	רק הצורות החוקיות.	כנ״ל
צורות ״ספרותיות״	מטופל רק אם נכלל במסגרת החוקים.	המילון מכיל את כל השפה העברית. קיים מידע על מקור המילה, כך שהאפליקציה יכולה להתעלם מצורות מסוימות.	כנ״ל

13.2 פונקציונליות דיוק הפגת העמימות

תחום	חוקה	מילון	שילוב
ניתוח הצורות הנכונות בלבד	חלקי. על פי החוקה	מלא	חוקה משופרת ע״י מאגר מידע
הפגת עמימות לקסיקלית	אין כלל טיפול לקסיקלי	במודל הבסיסי אין טיפול. מודל ההקשר מבוסס כנראה על אוטומט קצר הקשר הנותן פתרון לבעייה זו.	קיים מנגנון לקסיקלי.
בחירת הניתוח הלקסיקלי הבולט מקרב מספר ניתוחים	אין כלל טיפול לקסיקלי	אין מידע	ניתן להעזר במנגנון הסטטיסטי
ניתוח משפטים מורכבים	אין טיפול	כנראה שלא	כנראה שלא

13.3 מימוש

תחום	חוקה	מילון	שילוב
קלות המימוש	פשוט יחסית לאחרים.	דורש השקעה הן בפיתוח המודל והן בהקמת המילון (עשרות שנות אדם).	בינוני.
הרחבה ושילוב של שפות נוספות	דורש הקמת חוקים בלבד.	בכל מקרה של הרחבה לשפות נוספות ידרש תהליך של הקמת המילון. נראה שהוספת שפות שמיות תדרוש התאמות קלות במימוש המודל. לגבי שפות אחרות, נראה שמורכבות המודל תאפשר בקלות יחסית הוספת שפות ע״י התאמת המודל הדקדוקי. בהנחה שהדקדוק פשוט יותר.	הפתרון כללי. ניתן להרחיב את החוקה לשפות נוספות עם השקעה מעטה יחסית בבנית מאגר המידע.
שימוש בתשתית למימושים נוספים	רק לישומים המבצעים ניתוח מילה בודדת באופן גס יחסית.	כל שימוש מבוסס מילון.	למימושים המבצעים ניתוח מעודן יחסית וניתוח לקסיקלי.

13.4 אפקטיביות בתחום האחזור

תחום	חוקה	מילון	שילוב
RECALL	יחסית ליתר השיטות, ההתאמה בשיטה זו נמוכה יחסית. הטיפול הדקדוקי הוא כללי מדי ואיננו כולל טיפול במקרים יוצאי דופן.	התאמה גבוהה, היות והשיטה מטפלת בכל הצורות החוקיות של ערך מילוני, הרי שעבור מילים חוקיות ההתאמה מלאה. עבור מילים חדשות ההתאמה תהיה מינימלית.	ביחס לשיטות האחרות. הדיוק של הטיפול הדקדוקי טוב יותר משיטת החוקה, אך נמוכה יותר משיטת המילון. הטיפול כולל גם מילים שאינן במילון. לסיכום, השילוב של מילון סטטיסטי עם חוקה יביא להתאמה בינונית-גבוהה.
PRECISION	יחסית ליתר השיטות, ההתאמה בשיטה זו נמוכה יחסית ותגרום שליפה עם "רעש". הסיבה לכך היא כפולה. ראשית, הטיפול הדקדוקי הוא גס יחסית. שנית, המודל איננו כולל טיפול בעמימות לקסיקלית.	שימוש בשיטה זו יגרום ל"רעש" בשליפה של מילים חוקיות שיתכן שאינן רלוונטיות אך חוקיות בשפה. הוספת הרכיב של הטיפול בהקשר יאפשר לסנן את הפיתוחים הלשוניים שאינם רלוונטיים. התוצאה הסופית היא שההתאמה של תוצאות רלוונטיות מתוך תוצאות שנשלפו באמצעות ישום המבוסס על שיטה זו יהיה גבוה מאוד.	השילוב של הטיפול הדקדוקי עם המידע הסטטיסטי והטיפול בסינון פיתוחים הנובעים מעמימות לקסיקלית גורמים להתאמה גבוהה במדד ה-precision.

14. סיכום

בעבודה זו הוצגו שלוש שיטות עיקריות לטיפול במורפולוגיה עברית בישומי אחזור. מעיון בתוצאות ההשוואה עולה התמונה הבאה.

לשיטה הפועלת על פי חוקה יש יתרון בתחום המימוש. אך היא נחותה מהאחרות באופן משמעותי בהיבטים הפונקציונליים. הניתוח הלשוני המופעל על מילה בודדת נופל מיתר השיטות באורח משמעותי. קשה להגיע לתוצאות גבוהות בשיטה זו עבור טיפול במקרים יוצאי דופן ללא פגיעה בדיוק האחזור. באשר לבעיית העמימות, לשיטת החוקה אין מענה לבעיה זו.

לשיטת המילון יתרון בולט מול כל היתר בניתוח מילים הקיימות במילון. למעשה, אין לשיטה זו מתחרים בתחום הדיוק של ההטיות הלשוניות ויצירת אוסף הצורות החוקיות המתפתחות מערך מילוני. ההשקעה הדרושה לישום השיטה, שעיקרה בניית המילון, הינה של עשרות רבות של שנות אדם. השקעה זו גדולה עשרות מונים מהשיטות האחרות. בנוסף, חסרה בשיטה טיפול במילים שאינן מוכרות, למרות שהמילון מכיל גם סלנג.
הוספת רכיב הטיפול בהקשר נותנת מענה טוב לבעית העמימות על ידי האפשרות לסנן פיתוחים שהרלוונטיות שלהם נמוכה.

השיטה השלישית, משלבת את היתרונות העיקריים של שתי השיטות. הטיפול הלשוני הבסיסי דומה לשיטת החוקה. וההשוואה ביניהם בתחום זה תלויה בדיוק הדקדוקי שיושם בפועל. אולם, בניגוד לשיטת החוקה, הפיתוחים המתקבלים משלב זה עוברים סינון נוסף על בסיס חוקיות השפה. השלב האחרון, הוא סינון על בסיס מידע סטטיסטי. מבחינת הישום, ההשקעה הנדרשת בשיטה זו אמנם גבוהה יותר משיטת החוקה אך קטנה לאין ערוך משיטת המילון. היות והמידע הסטטיסטי-מילוני הוא צנוע הרבה יותר מהמילון של שוויקה.

לסיכום ההשוואה, נראה כי מתוך שלושת השיטות שהוצגו, לשיטת החוקה אין יתרונות בולטים.

באשר לשתי השיטות האחרות, המילונית והמשלבת, נראה כי לשיטה המשלבת יתרון בולט בתחום הישום. שכן לא נדרשות בה השקעות כה גדולות להקמת המילון.

כיום, היות והמילון כבר קיים, נראה שלשיטת המילון יש יתרון על השיטה המשלבת בתחום הדיוק של הניתוח הלשוני. אם כי קיימת בה הבעיתיות של טיפול במילים לא-מילוניות.
בתחום העמימות הלקסיקלית, נראה כי לצרוף של מסנן לקסיקלי עם מידע סטטיסטי המיושם בשיטת המשלבת יש יתרון על השיטה המילונית.

לסיכום, לכל אחת מהשיטות שהוצגו יתרונות שונים. נראה לכאורה שניתן ליצור סינתזה של השיטות בכדי למצות מכל אחת מהן את המירב. שילוב של ניתוח לשוני מדוייק המבוסס מילון יחד עם טיפול לשוני כללי עבור מילים חדשות יתן מענה מיטבי לטיפול במילה בודדת. באופן דומה, שילוב של ניתוח מדויק יחד עם סינון על בסיס חוקיות לשונית ומידע סטטיסטי עשויים להביא לתוצאות אופטימליות.

בנוסף לסינרגיה שהוצעה, יתכן שכיווני הפיתוח של הפגת העמימות יגיעו מניתוח יחידות מידע שלמות, מעבר לניתוח לקסיקלי של משפט בודד. ואפשר שיסייע בנושא מידע שאיננו קיים כיום כמו אינטונציה במערכות המשלבות דיבור.

פתחנו את העבודה במובאה מספר בראשית, "ויהי כל הארץ שפה אחת ודברים אחדים". שתיארה את הבהירות בתקשורת הבינאישית בתקופה שקדמה למגדל בבל. עולם המחשבים שהחל כ"מגדל בבל" טכנולוגי עם ריבוי שיטות יצוג וללא סטנדרטים אחידים צועד לקראת אחידות טכנית בעזרת סטנדרטים כמו Unicode, מעבר למערכות פתוחות המדברות זו עם זו ואימוץ טכנולוגיות מתחומים אחרים (DVD, CD, DAT). נסיים בתקווה שכלים לניתוח לשוני יסייעו ליצירת תקשורת ברורה יותר בין אנשים.

ביבליוגרפיה

[בנטור, 1992]

בנטור א', אנג'ל א', בן ארי-שגב ד', לביא א', "ניתוח ממוחשב של מילים עבריות", בלשנות חישובית עברית, משרד המדע והטכנולוגיה, 1992, עמודים 38-36.

[דגן, 1992]

עידו, ד', אלון, א', "גישה סטטיסטית רב-לשונית לפתרון בעיות רב-משמעות בשפה טבעית", בלשנות חישובית עברית, משרד המדע והטכנולוגיה, 1992, עמודים 145-139.

[הרץ, 1992]

הרץ, י', רמון, מ', "מיתון עמימות לקסיקלית ושימושים נוספים של אוטומט הקשר קצר", בלשנות חישובית עברית, משרד המדע והטכנולוגיה, 1992, עמודים 87-74.

[מט"ח, 2000]

אתר האינטרנט של פרוייקט "רב-מילים", המרכז לטכנולוגיה חינוכית, ת"א.
http://www.cet.ac.il/rav-milim/system.htm

[פנקס, 1985]

פנקס, ג'. "מערכת לינגואיסטית לאחזור מידע", הכנס הארצי ה20- לעיבוד נתונים, ירושלים, תשמ"ה, 1985, אילי"א, עמ' 342-329.

[Carmel, 1999]

Carmel, D., Maarek, Y., "***Morphological Disambiguation for Hebrew Search Systems***". In Proceedings of the 4th International Workshop, NGITS-99. Zikhron-Yaakov, Israel, July 1999. Lecture Notes in CS 1649, Springer. pp 312-325

[Choueka, 1978]

Choueka, Y., Attar, R., Dershowitz, N., Fraenkel, A.S, "***KEDMA - Linguistic Tools for Retrieval Systems***", Journal of Association for Computing Machinery, Vol 25 No. 1, 1978, pp. 52-66.

[Dewire, 1994]

Dewire, D. T., "***Text Managment***", McGraw-Hill Inc., NY, 1994.

[Ornan, 1997]
מתוך אתר האינטרנט של מכון ויצמן למדע, רחובות
http://www.weizmann.ac.il/home/comartin/ivrit/ansi.html

[Witten, 1994]

Witten, I.H., Moffat, A., Bell, T.C., "***Managing Gigabytes - Compressing and Indexing Documents and Images***",Van Nostrand Reinhold, NY, 1994.

אחזור מידע ומורפולוגיה של השפה הערבית

דרור קמיר

התמודדות עם שפות שמיות במסגרת עיבוד שפה טבעית (NLP)

דרור קמיר

מלינגו בע"מ – יישומי שפה טבעית www.melingo.com

הקדמה

השפות השמיות, בעיקר עברית וערבית, מציבות בעיות מורכבות בכל הקשור בשימוש במנועי-חיפוש, מנגנונים לאחזור מידע, ו-TTS. השימוש בשפות אלה במסגרת האינטרנט ומערכות ממוחשבות אחרות הולך וגובר, והפתרונות לחיפוש ואחזור מידע בשפות אלה נחוצים מאוד. חברת **מלינגו** פיתחה את מערכת "**מורפיקס**" שהיא מנגנון ייחודי לטיפול בטקסטים עבריים וערביים, שמשיג תוצאות טובות יותר מכל מערכת אחרת מבחינת ה-precision וה-recall.

המורפולוגיה הייחודית של השפות השמיות

הייחוד של השפות השמיות לעומת שפות אחרות נובע מהמורפולוגיה העשירה שלהן שפועלת באמצעות שורשים עיצוריים. הנטייה היא בלתי-רציפה (non-concatinative), כך שאי אפשר להבחין בגזע קבוע למילה לאורך פרדיגמת הנטייה. בהשוואה לשפות לא-שמיות, אפשר לומר שגזע-המילה משתנה כביכול במהלך הנטייה. בנוסף עשויות להצטרף לנטייה מוספיות (affixes)דוגמה: "התנגדתי" – שורש: ngd גזע: nagad מוספיות: **ti**-nagad-**hit**.
התופעות האלה ניכרות במיוחד בנטיית הריבוי בערבית המכונה "ריבוי שבור". דוגמה: دكان >> دكاكين (*plural*) dakakiin << (*singular*) dukkan.

השפות כוללות סמנים מורפולוגיים ותחביריים, המכונים "כינויים חבורים" ו"אותיות שימוש" (אפשר לראות בהם clitics). סמנים אלה משמשים כמילות-יחס, תוויות, כינויי גוף וסימוני יחסות שמצטרפים למילה ונכתבים כחלק ממנה. דוגמה: **בית**: **ה**בית, **ל**בית, בית**ם**, **ל**בית**ם** וכו'.
העברית והערבית כוללות כל אחת כ-70 מיליון מחרוזות חוקיות, כלומר 70 מיליון רצפים של אותיות שאפשר לקרוא כל אחד מהם כנטייה של מילה כלשהי בשפה. כל מילה בשפה יכולה להיכתב באלפי צורות אפשריות, בהתחשב במספר הנטיות שלה, ובאפשרות לצרף אליה אותיות שימוש.

המספר העצום של הצורות יוצר לעתים קרובות הומוגרפים - דמיון מקרי בין נטיות שנגזרות ממילים שונות לחלוטין. דוגמה: הצורה "מונות" נגזרת מ"להונות" או מ"למנות". לכך יש להוסיף את שיטת הכתיב שנמנעת מסימון תנועות (ראו להלן), כך שהרב-משמעות בשפות השמיות מגיע ל-50%, כלומר, כל מילה שנייה אפשר לקרוא בשני אופנים או יותר.

האורתוגרפיה של העברית והערבית

אף על-פי שעברית וערבית נכתבות באלפביתים שונים, שיטת הכתיב דומה מאוד. הכתיב מסמן בעיקר עיצורים, התנועות כמעט שאינן מסומנות. בעברית נהוגות שתי שיטות כתיב במקביל: כתיב מלא וכתיב חסר. הכתיב המלא מסמן תנועות במידה גדולה יותר מאשר הכתיב החסר, אולם סימוני התנועות הן אותיות שמסמנות גם עיצורים (א, ה, ו, י). זאת ועוד – שתי השיטות משמשות זו בצד זו, כשיש מספר גרסאות של שיטת הכתיב המלא. דוגמה: המילה *oznayim* נכתבת בארבע צורות: אזנים, אוזנים, אזניים, אוזניים. בערבית יש שיטת כתיב אחת, שבה מסומנות רק התנועות הארוכות, אולם, גם בה, האותיות המסמנות תנועות ארוכות מסמנות גם עיצורים (ا, و, ي). בנוסף, יש אותיות, בעיקר המזה, שמופיעות כסימנים שונים (אלוגרפים) בתפוצה משלימה (ء, أ, ؤ, ئ). התחלופה בין האלוגרפים מקשה על זיהוי גזעי המילים.

אין בערבית שיטה אחת לתעתוק מילים זרות ושמות זרים. שיטת התעתוק משתנה על-פי קונבנציות מקומיות וטעמו האישי של הכותב. כך שמה של מדינת אורגון בארה"ב יכול להיכתב ב-15 צורות שונות: أوريجون, أوريغون, أريجون, أريغون, أوريكون ועוד.

<u>אבחנה בין שתי רמות של מורפולוגיה</u>

היתרון המשמעותי שמתגלה בשיטת השורשים העיצוריים הוא האפשרות ליצור מנגנון חיפוש הפועל בשתי רמות: רמת מורפולוגיה נטייתית (inflectional morphology) ורמת מורפולוגיה גזירתית (derivational morphology). למשל, בעברית המילים: "חוק" ו"חקיקה" נגזרות מאותו השורש. אף על פי שהן מילים שונות, הן שייכות לשדה-סמנטי אחד, והבאתן יחד, כתוצאות של חיפוש אחד, עשויה להיות מועילה. שימוש בשורש העיצורי מייעל מאוד את יצירת הקשרים בין המילים. יש אמנם צורך באבחנה סמנטית כלשהי, כדי להפריד בין "חוק" ו"חקיקה" לבין "חקוק" שמשמעותו שונה לחלוטין, אולם הסמנטיקה נתמכת היטב על-ידי המורפולוגיה (באנגלית אין כמעט תמיכה כזו: law, legislation).

<u>פתרונות קיימים בלתי-מספקים</u>

מכל האמור לעיל, אפשר לראות שחיפוש פשוט הוא חסר ערך בעברית ובערבית. המורפולוגיה העשירה, והשיטה האורתוגרפית פוגמות הן ב-precision והן ב-recall. חיפוש הביטוי: "ערבי שירה" יכול להניב תוצאות כגון: "נעצר איש **ערבי שירה** לכל עבר". לעומת זאת טקסטים הכוללים את הביטויים: "ערב שירה", "ערבי השירה", "בערבי השירה" וכדומה, לא יובאו כלל. ניסיון לחפש על-פי גזעי-המילים אינו משפר בהרבה את התוצאות. חיפוש כזה יביא את המילים "סירים" ו"סירות" עבור אותה שאילתה, כיוון שבשתיהן אפשר לזהות את "סיר" כגזע-המילה. מאידך, המילה "חגיגה" לא תובא עבור השאילתה "חגג", כיוון שהגזע שלה הוא לכאורה "חגיג" ולא "חגג".

חיפוש על-פי הפרדת העיצורים ושחזור השורש פוגם בעיקר ב-precision. חיפוש כזה יביא את המילה "אסלה" בחיפוש שם העיר "אוסלו", כיוון שבשתיהן אפשר לבודד את העיצורים אס"ל, ולראות בהם את שורש המילים כביכול. זיהוי השורש שע"ר במילה "שיעור" יאפשר הבאת צורות כגון "שיעורים", "השיעור", אולם יגרום גם להבאת צורות לא רלוונטיות כגון "שערים". בנוסף לכך, יש מקרים רבים שבהם קשה לשחזר את השורש באמצעות אלגוריתם פשוט, למשל: מהצורה "מבוכה" אפשר להפיק את השורשים: מב"ך, בו"ך או בכ"י, מהמילה "מרכוז" את רכ"ז או מרכ"ז, מהמילה "קונן" את קו"ן, קי"ן או קנ"ן.

<u>הפתרון של "מורפיקס"</u>

מערכת "**מורפיקס**" מחלקת את כלל הצורות הלגיטימיות בעברית ובערבית ל"למות" (lemmas), שהן היחידות הבסיסיות הנחוצות בחיפוש יעיל. כל "למה" מזוהה באמצעות "מספר זהות", כדי למנוע טעויות שעשויות לנבוע מהאורתוגרפיה. הפתרון של מערכת מורפיקס כולל מספר מרכיבים המבטיחים שכל שאילתה תניב תוצאות הכוללות צורות מה"למה" שלה ולא מ"למות" אחרות שיש להן צורות דומות.

1. **אימות לקסיקלי** - מורפיקס פועלת על בסיס מילון. כל צורה שמופיעה בטקסט מנותחת ונבדקת מול לקסיקון שכולל את כל המילים השימושיות בשפה. כך, למשל, הצורה "סירות" תקושר רק ל"סירה" ולא ל"סיר", כיוון שהצורה "סירות" אינה קיימת בלקסיקון כריבוי של "סיר". במקרה שהאימות הלקסיקלי מאפשר כמה ניתוחים, המערכת תשקלל את הסבירות של כל אחד מהם, ותקבע מהו הסביר ביותר. המילון מתעדכן באופן שוטף ומקיף את כל המילים שעשויות להופיע בטקסט כתוב, כולל שמות-פרטיים, שמות מקומות, מילים לועזיות, מילים בשפה תת-תקנית וכדומה.
2. **הקשר** – המערכת של "מורפיקס" משתמשת ביחסים התחביריים בין המילים כדי לקבוע מהו הניתוח בעל הסבירות הגבוהה ביותר. קיים מאגר של חוקי תחביר שמאפשר לבחון את

היחסים בין הצורות השונות במשפט. כך הצורה "הטיל" תנותח כשם-עצם במשפט "הטיל שוגר לחלל", אך כפועל במשפט: "השר הטיל מסים חדשים". השאילתה "טיל" תביא, אם כן, את המשפט הראשון אבל לא את המשפט השני.

3. **קולוקציות –** קיים מאגר של קולוקציות שמאפשר לאתר צירופים קבועים של שתי מילים או יותר. השאילתה: "מחבל" לא תביא, אם כן, את הצירוף "מחבל הכביסה", כיוון ש"חבל כביסה" הוא צירוף מוכר בשפה.
4. **סאונדקס** - מורפיקס מפעילה אלגוריתם מתוחכם שמאפשר לאחד מילים זהות שנכתבו בשיטות כתיב שונות. הדבר מועיל במיוחד כשמדובר בתעתיקים של מילים זרות ושמות זרים לערבית. למשל, כל אחת מ-15 הצורות האפשריות לכתיבת השם "אורגון" עובר תהליך ניתוח שמשווה בין שיטות הכתיב השונות ומאמת את התוצאה בלקסיקון.

מערכת "**מורפיקס**" מאפשרת חיפוש צר לפי מורפולוגיה נטייתית וחיפוש רחב לפי מורפולוגיה גזירתית. חלק מה"למות" מקובצות יחד לקבוצות, על-פי השורש העיצורי שלהן ועל-פי השדה הסמנטי שלהן. בחיפוש רחב יובאו יחד המילים "לשתוק", "שתיקה", "להשתיק", "להשתתק" אבל לא יובאו המילים "שיתק", "שיתוק".

"**מורפיקס**" מאפשרת חיפוש לפי **תזאורוס**, כלומר חיפוש של מילים נרדפות למילה המופיעה בשאילתה. החיפוש לפי תזאורוס מגשר על הבדלים בין אזורים שונים בעולם הערבי שבהם מקובל לעתים קרובות מינוח שונה. כמו כן, הוא מסייע במקרים שבהם תחדיש לשוני מחליף בהדרגה מילה שאולה, למשל "קלטת" מול "קסטה" בעברית, ו-هاتف מול تلفون בערבית.
כמו כן, "**מורפיקס**" מאפשרת חיפוש טרנס-לינגוויסטי אנגלי-ערבי. המערכת מקבלת שאילתה באנגלית, ומחזירה את כל הצורות של כל התרגומים האפשריים לערבית. אם השאילתה היא תעתיק של שם ערבי לאנגלית, המערכת תחשב את הצורות הערביות שיכולות להתאים לתעתיק, ותאמת אותן בלקסיקון. כיוון שאין שיטת תעתיק אחידה מערבית לאנגלית או לאותיות לטיניות בכלל, אפשר למצוא עשרות תעתיקים לחלק מהשמות הערביים הנפוצים (למשל: Muhammad, Mohammad, Mohammed, Mehammed, Mouhammad ועוד). החיפוש הטרנס-לינגוויסטי מאפשר איתור שמות פרטיים בתוך טקסטים ערביים על-פי התעתיקים הנפוצים לאנגלית.

לקריאה נוספת:
Kamir, Dror, Naama Soreq and Yoni Neeman (2002). “A Comprehensive NLP System for Modern Standard Arabic and Modern Hebrew”, *Computational Approaches to Semitic Languages - Proceedings of the Workshop*, ACL-02 Workshop on Computational Approaches to Semitic Languages, pp. 76-84.

מנועי אחזור טקסט בעברית - רשימת ספקים (גירסה 3.2004)

עפר דרורי

מנועי אחזור טקסט בעברית - רשימת ספקים
גרסה 3.2004

[עפר דרורי](#)

offerd@cc.huji.ac.il

מבוא
מערכות מידע בעבר טיפלו בעיקר בניהול רשומות בתוך בסיסי נתונים כאשר רוב המידע ברשומות היה מידע מפורמט בשדות נתונים בעלי אופי מוגדר מראש (הן בגודל שדות המידע והן בפורמט שלהם). מזה כמה שנים מערכות מידע נדרשות לטפל גם במידע שאיננו מפורמט כמו טקסטים, תמונות, קבצי קול ועוד. גם בהווה בו קיימים מדיות שונות עדיין מרכיב הטקסטים במערכות המידע הוא גדול ביותר. מכיוון שטיפול בטקסטים הוא משימה מחשבית מורכבת מתקיים בתחום הנוהג שמפתחי מערכות אינם מפתחים מנועי אחזור טקסט למערכות המידע שהם כותבים בדומה לכך שלא נהוג לפתח תכונות של עיבוד טקסט המתקיים במעבדי תמלילים.

"עברית שפה קשה" אמר המשורר ובכל הקשור לטיפול ממוחשב בשפה העברית על אחת כמה. נהוג לדרג את השפות בעולם על פי הקושי הנדרש בטיפול ממוחשב בהן. בתחתית הסולם נמצאת השפה הסינית שבה אין נטיות ואין רב משמעות למילים. אחרי הסינית מבחינת הסיבוכיות נמצאת האנגלית, אחריה צרפתית כאשר העברית והערבית נחשבות כשפות הקשות ביותר לטיפול ממוחשב מכיון שהן מכילות הטיות רבות, מורפולוגיה מורכבת וריבוי משמעויות.
את התכונות הנדרשות בשפה העברית ניתן לחלק לטיפול בטקסט ולטיפול בממשק (במידה והוא מסופק עם המוצר).
טיפול בטקסט מתייחס לתכונות כמו: כיווניות שפה, מורפולוגיה (שהיא ייחודית לשפה), אחזור על פי שורש מילה (השונה מהותית בשפה העברית משפות לועזיות אחרות), צליל (סאונדקס), גידומים (אשר יש להם משמעות רבה יותר באנגלית מעברית), טיפול בתזאורוס המותאם לשפה ועוד.
טיפול בממשק מתייחס לשפת התפריטים, כיווניות השפה המוקלדת בעת ביצוע שאילתת החיפוש, להצגת המידע, לעזרה המקוונת ועוד. כאמור יש להתייחס למרכיב זה כאשר המוצר כולל ממשק.

מטרת מסמך זה להציג את רשימת הספקים והמוצרים הקיימים בתחום, התומכים בשפה העברית וניתנים להשגה בארץ. מסמך זה נילווה למסמך "קריטריונים לבחירת מנוע אחזור טקסט" ואשר יכול לסייע בתהליך בחירת מנוע מסחרי מסוים מתוך רשימה של מספר מנועים. המסמך עצמו נמצא ב- http://shum.huji.ac.il/~offerd/papers/drori112002h.pdf

להלן רשימת הספקים והמוצרים הנתמכים בארץ וכוללים טיפול מסוים בשפה העברית. כפי שנאמר הטיפול בעברית יכול להיות בכמה רמות ועל הארגון הבוחר את המוצר לתת את הדעת לנושא זה כמו לתכונות האחרות של המוצר. הרשימה כוללת מוצרים שניתן להפעילם על פלטפורמות שונות ושאינם מוגבלים לעבודה מול בסיס נתונים מסחרי אחד.

רשימת המנועים מעודכנת ל- 3.2004 באדיבות היצרנים והנציגים, אם הנך נציג מוצר התומך באחזור טקסט בעברית או אם אתה משתמש ומכיר מוצר כזה אנא העבר לי פרטיו כדי שאוכל לשבץ אותו בטבלה לתועלת הציבור המתעניין בתחום. ניתן להעביר את הפרטים באמצעות דוא"ל ל- offerd@cc.huji.ac.il. עדכונים למוצרים עצמם, גרסאות או פרטים

מזהים אחרים יתקבלו בברכה.

מסמך זה הוא גרסה מעודכנת שלישית למסמך המקורי שיצא לאור בשנת 2002.

פרטים מזהים של הספקים

שם המוצר	XRS	InetrText
שם קודם		TQL
גרסה נוכחית	4.3	4.1
שם החברה המפתחת	2001 שרותי מערכות ומחשבים בע"מ	Inter Dbtec
כתובת החברה	דוב גרונר 12 הרצליה פיתוח 46723	יוני נתניהו 3ב אור יהודה
שם הנציגות בארץ	כנ"ל	SPL Software
כתובת הנציגות	כנ"ל	כנ"ל
שם איש הקשר בארץ	אסתר אמסלם	ראובן איבגי
טלפון איש קשר	09-9511225	03-5388331
פקס איש קשר	09-9511226	03-5335511
דואר אלקטרוני	esthera@int2001.co.il	reuven_ivgi@splsoftware.com
אתר אינטרנט של המוצר	www.int2001.co.il	www.interdbtec.com
הערות		
מעודכן לתאריך	3.2004	3.2004

שם המוצר	Verity	RetrievalWare
שם קודם		
גרסה נוכחית	5.0.1	8.0
שם החברה המפתחת	Verity	Convera (לשעבר Excalibur)
כתובת החברה	ארה"ב	ארה"ב
שם הנציגות בארץ	מטריקס - חטיבת מוצרים	HP ישראל
כתובת הנציגות	שד' הגלים 3 ת.ד. 2016 הרצליה 46120	דפנה 9, רעננה
שם איש הקשר בארץ	יורם זהבי	עמיחי רימר
טלפון איש קשר	09-9598889	09-7623408
פקס איש קשר	09-9598822	09-7427675
דואר אלקטרוני	yoramz@matrix.co.il	amichay.rymer@hp.com
אתר אינטרנט של המוצר	www.verity.com	www.converta.com
הערות		
מעודכן לתאריך	3.2004	3.2004

שם המוצר	WizDoc	DTSearch
שם קודם		
גרסה נוכחית	1.3	6.3.1
שם החברה המפתחת	חשבשבת - WizSoft	DTSearch
כתובת החברה	בית הלל 3, תל אביב	ארה"ב
שם הנציגות בארץ	כנ"ל	TDS
כתובת הנציגות	כנ"ל	לחי 31, בני ברק
שם איש הקשר בארץ	ארז מזרחי	יגאל רחמים
טלפון איש קשר	03-5631919	03-5782818
פקס איש קשר	03-5611864	03-5782820
דואר אלקטרוני	erez@wizsoft.com	yigal@tds.co.il
אתר אינטרנט של המוצר	www.wizsoft.com	www.dtsearch.com www.tds.co.il
הערות		
מעודכן לתאריך	3.2004	3.2004

שם המוצר	Contex	Fast DataSearch
שם קודם		
גרסה נוכחית	3.1	4.0
שם החברה המפתחת	LiveLinx	Fast Search & Transfer
כתובת החברה	בית רמפא הר חוצבים ים	נורבגיה וארה"ב
שם הנציגות בארץ	כנ"ל	Broad-net מקבוצת One1
כתובת הנציגות	כנ"ל	ברזל 21 רמת החייל ת"א
שם איש הקשר בארץ	יניב שושני	עוזי אחיטוב
טלפון איש קשר	02-5328580/102 053-474826	051-323300 03-7677900
פקס איש קשר	02-5328320	03-7677901
דואר אלקטרוני	yanivs@livelinx.com	uachituv@one1.co.il
אתר אינטרנט של המוצר	www.livelinx.com	www.fastsearch.com www.alltheweb.com
הערות		
מעודכן לתאריך	3.2004	3.2004

שם המוצר	**Flair**
שם קודם	
גרסה נוכחית	
שם החברה המפתחת	נס-טכנולוגיות (לשעבר קונטהל)
כתובת החברה	עתידים בנין 1, תל אביב
שם הנציגות בארץ	כנ"ל
כתובת הנציגות	כנ"ל
שם איש הקשר בארץ	עפרה פרנקל
טלפון איש קשר	03-7673565
פקס איש קשר	03-6497713
דואר אלקטרוני	ofra.fraenkel@ness.com
אתר אינטרנט של המוצר	www.ness.com
הערות	רק במסגרת פרוייקטים
מעודכן לתאריך	3.2004

מנוע מורפולוגי (לעברית וערבית)

שם המוצר	**מורפיקס**
שם קודם	
גרסה נוכחית	ראה הערות
שם החברה המפתחת	מלינגו
כתובת החברה	תוצרת הארץ 16 תל אביב
שם הנציגות בארץ	כנ"ל
כתובת הנציגות	כנ"ל
שם איש הקשר בארץ	ליאורה ירדני
טלפון איש קשר	03-6070420
פקס איש קשר	03-6070401
דואר אלקטרוני	lioray@melingo.com
אתר אינטרנט של המוצר	www.morfix.co.il
הערות	גרסאות למנועים השונים: RetrievalWare, Verity, Fast
מעודכן לתאריך	3.2004

עפר דרורי

קריטריונים לבחירת מנוע אחזור טקסט

גרסה 3 – 3.2004

עפר דרורי

offerd@cc.huji.ac.il

מבוא

מטרת מסמך לזה להגדיר טבלת קריטריונים לצורך השוואה בין מנועי חיפוש לאחזור טקסט. עם הגידול במאגרי המידע ובכמויות המידע המילולי בהן (בעיקר טקסטים שאינם מפורמטים) עולה הצורך בכלים / מוצרים אשר ינהלו את המידע הרב הזה. ניהול טקסטים חופשיים בצורה איכותית, בעיקר בכל הקשור לאחזור המידע מהם, הוא משימה מחשבית שאינה קלה כלל ועיקר. בגלל מורכבות המשימה מחד והצורך בפתרון בעל ביצועים סבירים מאידך נוצרה בתחום זה התמחות אשר מסופקת על ידי מספר רב של חברות בעולם. נפיצותם של מערכות לאחזור טקסט ומנועי חיפוש גדלה רבות יחד עם התפתחות רשת האינטרנט והצורך לאיתור מידע מהרשת. ככל שהמאגרים גדולים יותר וככל שהמידע המוזרם לתוכם לא עובר תהליכי סינון ובקרה, כך גדלה חשיבותם של המנועים ובמיוחד תכונותיהם. במאגרים מקצועיים בהם הבקרה על המסמכים רבה, הצורך בתכונות משוכללות של המנוע קטנה במידה מסוימת אף שגם במאגרים מסוג זה קטלוג החומר ומפתוחו לא תמיד עונים על כל הצרכים. בנוסף אי אפשר להתעלם שגם במאגרים המקצועיים נעשים ניסיונות "לעיגול פינות" בגלל העלויות הגדולות הכרוכות בבקרה זו.

המסמך כאמור, מרכז קריטריונים רבים לצורך בדיקה והשוואה בין מוצרים שונים והוא מהווה כלי עזר לבחירה של מוצר לארגון על פי הצרכים הארגוניים שנקבעו מראש. הטבלה כוללת התייחסות למספר נושאים:

נתוני זיהוי של המוצר, החברה המפתחת והנציגות בארץ אם קיימת
תכונות המוצר
טכנולוגיה של המוצר
מידע על הספק וניסיונו
ביצועים של המוצר
מחירי המוצר בהתייחס לתצורות השונות

מומלץ להעביר את הטבלה המצורפת להתייחסות הספקים השונים. יש להחליט כמובן על השקלול המתאים של התכונות השונות עפ"י מודל דלפי לברירת חלופות.

פורמט Word של המסמך נמצא בכתובת
http://shum.huji.ac.il/~offerd/papers/search-engine-critrion.doc

פורמט PDF של המסמך נמצא בכתובת
http://shum.huji.ac.il/~offerd/papers/drori032004h-2.pdf

מסמך המפרט את ספקי מנועי האחזור התומכים בעברית נמצא בכתובת
http://shum.huji.ac.il/~offerd/papers/drori032004h.pdf

מידע נוסף על תכונות המנועים, אחזור טקסט ועוד נמצא באתר קבוצת העניין אחזור טקסט – SIGTRS
http://sigtrs.huji.ac.il

פרטים מזהים

שם המוצר	
שם קודם של המוצר	
מספר גרסה נוכחית	
שם החברה המפתחת	
כתובת החברה	
שם הנציגות בארץ	
כתובת הנציגות	
שם איש הקשר בארץ	
טלפון איש קשר	
פקס איש קשר	
דואר אלקטרוני	
אתר אינטרנט של המוצר	

תכונות

קריטריון	הסבר	התייחסות הספק
אחזור על מסמכי טקסט	תכונה בסיסית, בהתייחס לטקסט חופשי	
אחזור על מידע מפורמט	היכולת לבצע אחזור על שדות מידע רגילים	
אחזור לשדות מפורמטים בתוך מסמך טקסט	כמו תאריך, מחבר וכו'	
אופרטורים בולאנים	AND, NOT, OR, <, >, <=, >=, שימוש ב- ( ) לביטויים מורכבים	
אופרטורים מטריים (מטריקה)	"מילה" AND "מילה שנייה" במרחק X מילים, באותו משפט, באותה פסקה וכו'	
אחזור לשפה עברית	הכוונה לטיפול מיוחד בשפה ולא לאחזור על בסיס ייצוג האותיות העבריות במאגר	
אחזור לשפה אנגלית		
אחזור ל- 2 השפות במעורב	באותו מסמך	
אחזור לשפות נוספות		
מילון מורפולוגי כחלק מהמוצר	האם קיים כזה ואם כן מהן תכונותיו	
שילוב מילון מורפולוגי חיצוני	האם השילוב אפשרי, אם כן ציין איזה מילון (כמו מורפיקס של מלינגו)	
תמיכה בטבלאות תזאורוס במוצר	האם קיימת תשתית לשימוש בטבלאות תזאורוס חיצוניות, האם המוצר כולל תזאורוס משלו, אם כן לאיזה תחום ובאיזה שפה	
ניהול תזאורוס במוצר	במידה ואנו רוצים לייצר את המילון לבד, האם יש תמיכה לשימוש בטבלה ריקה שתוזן ע"י המשתמש	
ניהול תזאורוס היררכי	האם קיימת תשתית במוצר לתזאורוס היררכי	
הצגת תוצאת החיפוש ע"י המוצר	האם המוצר כולל ממשק משתמש להצגת תוצאות החיפוש, אם כן באיזו שיטה: רק כותרות, תחילת מסמך, משפטיים רלוונטיים לחיפוש וכו' – יש לפרט	
הצגת שאילתת החיפוש ע"י המוצר	האם המוצר כולל ממשק משתמש לביצוע שאילתת החיפוש	
הדגשת מילים המקיימים את תנאי החיפוש	האם המוצר כולל תמיכה בהדגשת מילים המקיימות את תנאי החיפוש במסמכים שאותרו	
ביצוע שאילתות על שאילתות	יצירת סטים של תשובות וביצוע אחזור נוסף עליהם	

אחזור על מספר מאגרים במקביל	האם אפשרי ואם כן האם גם בפורמטים שונים של המאגרים	

קריטריון	הסבר	התייחסות הספק
אחזור מדויק בלבד	האם קיים בלי הרחבות אוטומטיות	
אחזור פונטי	האם קיים	
הרחבות אחזור לחלקי מילים (Wildcard)	ראשיות, סופיות ואמצעיות אפשרות להחלפת תווים בסימן "*"	
ניהול אינדקסים במוצר	האם נעשה במוצר עצמו, האם מתבסס על בסיסי נתונים חיצוניים, תיאור שיטת עבודה	
ניהול הרשאות גישה לקטעים במאגר	האם יש תמיכה להרשאות שונות על קטעים במאגר	
גיבוי ושחזור במוצר	האם כולל מנגנון גיבוי ושחזור עצמי או שמתבסס על גיבוי ושחזור חיצוני של כל סביבת העבודה	
תמיכה באחזור מסמכי טקסט ASCII	האם קיים, מגבלות אם יש נא לציין	
תמיכה באחזור מסמכי טקסט מסוג WORD	להתייחס לאיזה גרסאות WORD, הכוונה לאחזור ישיר ממסמכי WORD ללא הסבתם	
תמיכה באחזור מסמכי טקסט מסוג HTML	האם קיים	
תמיכה באחזור מסמכי טקסט HTML בעברית לוגית וחזותית	האם קיים, ציין מגבלות לגבי לוגית וחזותית	
אחזור מתוך מסמכי Excel	האם קיים, הכוונה לאחזור ישיר ממסמכי Excel	
אחזור מתוך מסמכי PDF	האם קיים, ישירות מול קבצי אקרובט PDF	
אחזור מתוך מסמכי PS	האם קיים, ישירות מול מסמכי Post Script	
תרגום פורמטים	האם המוצר כולל תרגום פורמטים שונים לטקסט נקי, אם כן פרט אילו	
אחזור לשורשים בשפה העברית והאנגלית	האם קיים	
אפשרות להוספת מסמכים בצורה מקוונת	מתוך מערכת O.L., הכוונה למנגנון המאפשר הוספה מיידית של מסמך וביצוע אחזור מיידי למסמך במסגרת המוצר ולא להפעלת אצווה כל מספר שניות או דקות	
תמיכה בטיפול בטבלאות	אחזור וניהול המידע כאשר הוא בטבלאות	
דרוג (Ranking) תשובות	האם המוצר כולל מנגנון לדירוג רשימת התוצאות, פרט אלו אלגוריתמים קיימים לדירוג	

קריטריון	הסבר	התייחסות הספק
מנגנונים לסיווג	האם המוצר כולל מנגנוני סיווג כמו קטגוריזציה, אשכולות ועוד. נא לפרט גם בהתייחס לשפה	

טכנולוגיה

קריטריון	הסבר	התייחסות הספק
תמיכה בשרת/לקוח	האם קיימת	
תמיכה בלקוח תחת חלונות XP ,2000, 98 וכו' דפדפנים לסוגיהם	התייחס לגרסאות התומכות בסביבות העבודה השונות, לגבי דפדפנים התייחס לגרסאות השונות, אם ידועות בעיות בנושא - נא ציין	
תמיכה בשרתי NT, UNIX	האם קיימת	
קישור לבסיסי נתונים מסוג SQLServer , אורקל	האם קיימת תמיכה למידע המאוחסן בבסיסי הנתונים הנ"ל, אם קיימת תמיכה בבסיסי נתונים מקומיים אחרים נא ציין	
קישור ל- Exchange	האם קיים בכל הקשור לאחזור מידע המוטמע בשרתי Exchange	
קישור ל- Notes	האם קיים בכל הקשור לאחזור מידע המוטמע בשרתי Domino	
ממשקים מסוג RPC, OLE, Java, ACTIVEX	ציין אילו קיימים ואם קיימים נוספים נא ציין	
ממשק API	האם קיים ממשק API המאפשר ביצוע של כל התכונות מתוך תוכנה אחרת חיצונית	
קישור לשרת אינטרנט IIS	האם קיים	
קישור לשרת אינטרנט של אורקל	האם קיים	
קישור לבסיס נתונים ב- MF כאשר אינדקסים נמצאים ב- MF או בשרת	הכוונה למצב בו המידע המיועד לאחזור מאוחסן בבסיס נתונים מרכזי ב- M.F. כדוגמת ADABAS והאינדקסים לאחזור נמצאים בשרת או ב- M.F. (נא ציין במה המוצר תומך)	
תמיכה בריבוי אינדקסים	מה המספר המקסימאלי של אינדקסים שניתן להקים ולנהל?	
אינדוקס ישירות מבסיס הנתונים	האם מתבצע אינדוקס אוטומטי עם עדכון בסיס הנתונים?	
שמירת האינדקס בבסיס הנתונים	האם יש אפשרות לכלול את האינדקסים בבסיס הנתונים עצמו?	
תמיכה בבסיס נתונים מעל 5 מיליון מסמכים	האם קיימת	

ביצועים

קריטריון	הסבר	התייחסות הספק
זמן טעינה של 100,000 מסמכים	אינדוקס של 100,000 רשומות חדשות, נא ציין ביחס לשרת סטנדרטי (ציין חוזק השרת, מספר CPU וכו')	
זמן טעינה של 10,000 מסמכים	כנ"ל לגבי 10,000	
זמן תגובה לחיפוש ביחס לגודל המאגר	עפ"י מבחני ביצועים של המוצר (נא ציין המבחן)	
נפח אינדקס ביחס לגודל המאגר	בזמן נתון ועם גידול המאגר, נא לצרף נתונים	

מידע על הספק וניסיונו

קריטריון	הסבר	התייחסות הספק
ניסיון בשנים בפיתוח מנוע אחזור טקסט		
מספר התקנות בארץ		
מספר התקנות בחו"ל		
כמות התקנות למאגרים מעל 5 מיליון מסמכים		
כמות התקנות מעל מיליון מסמכים		
כמות התקנות מעל 100,000 מסמכים		
כמות התקנות בסביבת אינטרנט/אינטרה-נט		
כמות התקנות המשלבת בסיס נתונים ארגוני על PC	נא לציין את סוג בסיס הנתונים: אורקל, SQL-Server וכו'	
כמות התקנות המשלבת בסיס נתונים ארגוני על MF	נא לציין את סוג בסיס הנתונים: אדבס, DB2 וכו'	
כמות התקנות עם Exchange		
כמות התקנות עם Notes		
סוגי תמיכה	נא לפרט: תמיכה טלפונית או אחרת	
פרק זמן בין גרסה לגרסה		
מחויבות החברה למוצר	האם מוגבלת בזמן, האם מרכז עיסוק החברה בתחום, תוכניות לעתיד בהקשר למוצר	
רשימת לקוחות	עפ"י הקריטריונים השונים: לקוחות עם בסיס נתונים אדבס, DB2 אחר לקוחות עם בסיס נתונים אורקל , SQL-SERVER וכו'.	
מספר שנים שהמוצר הנוכחי מותקן אצל לקוחות		
השתתפות במכרזים או בחירות קודמות	אם המוצר השתתף במכרזים או בחירה אחרת ונבחר, נא ציין את הגורם הבוחר, שנת הבחירה ואת המוצרים שמולם עמד לתחרות	

מחירים

קריטריון	הסבר	התייחסות הספק
מחיר המוצר	נא להתייחס לגרסאות שונות של המוצר (אם קיימות) בהתייחס למספר שרתים, לאתר, שיטות רישוי שונות	
מחיר אחזקה לשנה לאחר תקופת האחריות	נא לציין את תקופת האחריות	

ייצור אוטומטי של תיזאורי ומילונים דו לשוניים

איריס ארד

ייצור אוטומטי של תיזאורי ומילונים דו-לשוניים

איריס ארד

מילות מפתח
מילונים דו-לשוניים, תזאורי, בלשנות סטטיסטית, קורפוסים, סיווג מילים.

תקציר
המאמר מתאר שיטות סטטיסטיות להפקה אוטומטית של ידע בלשני מקורפוס גדול של טקסטים. החלק הראשון של המאמר מתאר כיצד טכניקות עיבוד פשוטות, בהן חלוקת הקורפוס למשפטים, יצירת רשימות KWIC וספירת שכיחויות מאפשרות להפיק מילון דו-לשוני מקורפוסים בשפות מקבילות. החלק השני של המאמר מתאר כיצד הפעלת אותן טכניקות על קורפוס בשפה אחת בלבד מאפשרת לבנות תשתית לתזאורוס המשמש לאחזור טקסט.

מבוא
מערכות לאחזור טקסט נעזרות בידע בלשני לעיבוד ואחזור טקסטים. הידע הבלשני מתבטא בשימוש בכלים כגון מילונים, תזאורי, כללים מורפולוגיים וכו'. מחקרים בתחום אחזור טקסט מקדישים תשומת לב מעטה לבנייתם ותחזוקתם של כלים אלו. בנייה ותחזוקה של כלים בלשניים היא באחריותם של בלשנים מקצועיים המבצעים עבודה ידנית, איטית ויקרה. לעיתים בוחרים מפתחי מערכות לאחזור טקסט באופציה של קניית הכלים הבלשניים מגורם חיצוני. קניית כלים חיצוניים דורשת מאמץ טכני ואינטלקטואלי כדי להתאים את התוכנה ואת הידע הבלשני האצור בה לצרכי המערכת הספציפית.

מאמר זה דן בבניית מילונים דו-לשוניים ותזאורי עבור מערכות לאחזור טקסט. המאמר מתאר כיצד ניתן, באמצעים פשוטים ובמאמץ קטן יחסית, לייצר בצורה אוטומטית מילונים דו-לשוניים ותשתית לתזאורי על ידי הפעלת טכניקות סטטיסטיות על גופים גדולים של טקסטים. התוצאות המופקות על ידי השימוש בטכניקות המתוארות ניתנות לשימוש כמו שהן, או שניתן לשפרן על ידי התערבות אנושית. המאמר הוא תקציר של פרקים מתוך עבודת דוקטורט שנערכה במכון הטכנולוגי של אוניברסיטת מנצ'סטר שבאנגליה (UMIST) בין השנים 1989-1991.

רקע
המחקר שבמסגרתו פותחו השיטות המתוארות להלן, נערך בתחום התרגום הממוחשב. בשלב הראשון פותחה מערכת BITS לתרגום ביביליוגרפיות מעברית לאנגלית (Arad 1991). המערכת נזקקה לידע בלשני על מנת לפעול. באותה תקופה החל להתעורר עניין מחודש בבלשנות סטטיסטית, תחום שהוזנח וזכה לתשומת לב מועטה בלבד במשך תקופה ארוכה.[1] ראוי במיוחד לציון ניסיון שנערך במעבדת המחקר של IBM בג'ורג'טאון ארה"ב לייצר מילון אנגלי-צרפתי מ Hansard Corpus המכיל את הפרוטוקולים של הפרלמנט הקנדי (Brown et al. 1988). ניסיון זה שימש השראה לשלב השני של המחקר שבו הוחלט לנסות לייצר את הכלים הבלשניים הנחוצים למערכת BITS בצורה אוטומטית מקורפוס גדול של טקסטים. המחקר נוקט במכוון בגישה קיצונית הדורשת שהתוצאות יוזנו ישירות למערכת התרגום ללא התערבות אנושית. מאחר ולא ניתן היה להשיג קורפוס בגודל הדרוש לביצוע ניתוחים סטטיסטיים משמעותיים, נעשה שימוש בשיטות quasi-statistical ושכל ישר.

מערכת BITS נזקקה למילון דו-לשוני שבעזרתו תורגמו המילים במשפט המקור לשפת המטרה ולכללי תחביר שבעזרתם סודרו המילים המתורגמות כך שנוצר משפט משמעותי ותקין מבחינה תחבירית בשפת המטרה. תהליך ייצור המילונים והכללים היה מעניין כשלעצמו, ובהמשך התברר שניתן לנצל את הניסיון ואת הנתונים שנצברו ולהפעיל את אותן השיטות כדי לסייע בבניית מערכות אחזור טקסט.

---

[1] ניסיונות בבלשנות סטטיסטית החלו כבר בשנות החמישים של המאה העשרים. ראוי במיוחד לציון מחקרו של Zipf (1949). עד שנות השמונים של המאה העשרים נערכו רק ניסיונות בודדים שלא קידמו את המחקר בתחום. עניין מחודש בבלשנות סטטיסטית החל להתעורר עם התפתחות רשתות התקשורת וכניסת ה BITNET ואחריה ה INTERNET שהציעו קורפוסים גדולים וזמינים.

ייצור אוטומטי של מילונים דו-לשוניים

מילון דו-לשוני מורכב מרשימה של מילים וביטויים בשפת מקור, כאשר לכל ערך ברשימה מוצמד תרגום אחד או יותר בשפת מטרה. למען הפשטות, ומכיוון שמדובר בתחום ידע מסויים ומוגבל שבו מנוטרלת העמימות של מרבית המילים בטקסט, נניח שלכל ערך בשפת המקור יש תרגום אחד בלבד בשפת המטרה ונתעלם בשלב הראשון מהבעיות שיוצרים הומוגראפים.

הכנת המילון הדו-לשוני נחשבת לעבודה טכנית אנושית. לאנשים האחראים על הכנת המילון ידע מקיף בשפת המקור וגם בשפת המטרה, ובדרך כלל יש להם גישה לטקסט שעבורו נבנה המילון. הם נדרשים לבחור ולהחליט אילו תרגומים יכללו במילון ואילו תרגומים יושמטו. התוצאה (האפשרית) של בחירות והחלטות אלה היא שהמילון ישקף את הרקע התרבותי של יוצריו ויסבול מהטייה. ייצור המילון בצורה אוטומטית (או לפחות חצי אוטומטית) יבטיח תוצאות אובייקטיביות.

בשנת 1988 בצעו Brown et al. ניסיון ראשון לייצר מילון דו-לשוני בצורה אוטומטית. הם השתמשו ב Hansard Corpus המכיל את הפרוטוקולים של הפרלמנט הקנדי באנגלית ובצרפתית. הקורפוס (יותר מ 1,000,000 מילים שוטפות) חולק למשפטים מקבילים כך שידוע כי התרגום של משפט S_e באנגלית הוא משפט S_f בצרפתית. לאחר מכן נבנה אינדקס של המילים בקורפוס האנגלי שבו עבור כל מילה בקורפוס נרשמו כל המשפטים בהם היא מופיעה. עבור כל רשימת משפטים נבנתה רשימת המשפטים המקבילים בצרפתית. כל אחת מרשימות המשפטים בצרפתית חולקה למילים והמילים מוינו וסודרו לפי שכיחותן ברשימת המילים. נקבע כי המילה השכיחה ביותר היא התרגום של המילה בשפת המקור.

בעיות התעוררו בגלל מילות יחס, שלא ניתן היה למחוק מרשימת המילים, אך שכיחותן הגבוהה הפריעה לספירת המילים ושיבשה את התוצאות. Brown et al. אינם מספקים הערכה כמותית של תוצאות הניסוי שלהם, אך הדוגמאות שהם מביאים מעידות על אחוזי הצלחה גבוהים ועל יצירת מילון שימושי ומדוייק.

לשיטה הנ"ל מספר חסרונות שהגדול בהם הוא הצורך בקורפוס דו-לשוני גדול מאוד המכיל מיליוני מילים רצות. חיסרון נוסף שהשיטה מתעלמת מביטויים ופועלת תחת ההנחה שמילה בשפת המקור מתורגמת למילה אחת בלבד בשפת המטרה ולכן היא אינה מסוגלת לזהות שהמילה העברית *רמזור* תתורגם לביטוי האנגלי *traffic light*.

האלגוריתם הבא לייצור מילון דו-לשוני נגזר מתוך האלגוריתם המתואר למעלה, אך הוא מנסה לתת מענה לבעיות שעוררה השיטה המקורית: ניתן להפעילו ברמת דיוק גבוהה יחסית על קורפוסים קטנים יחסית (בין 30 ל 60 אלף מילים), והוא מציע דרך לטפל בביטויים, בהם ביטויים לא רציפים.

השלבים הראשונים ביצור המילון זהים לשלבים הראשונים של האלגוריתם המקורי: גם כאן נחוץ קורפוס דו-לשוני המחולק למשפטים מקבילים. גם כאן נבנה אינדקס של המילים בשפת המקור, כאשר עבור כל מילה בקורפוס נרשמים כל המשפטים בהם היא מופיעה. גם כאן עבור כל רשימת משפטים בשפת המקור נבנית רשימת המשפטים המקבילים בשפת המטרה.

אולם כאן מסתיים הדמיון לאלגוריתם המקורי. במקום לבנות רשימת מילים ולחפש את המילה השכיחה ביותר בשפת המטרה, משתמש האלגוריתם שלנו בפעולת חיתוך: האלגוריתם מוצא את כל המילים המשותפות לזוג המשפטים הראשון ברשימה והתוצאה נשמרת בקבוצת חיתוך. ממשיכים וחותכים את קבוצת החיתוך עם המשפט הבא ברשימה וכך הלאה עד לסיום רשימת המשפטים. בצורה זו מספר התרגומים האפשריים של המילה הולך וקטן במקום לגדול. אם חיתוך מסוים מייצר קבוצת חיתוך ריקה, ניתן לחזור צעד אחד אחורה ולהתעלם מהמשפט שיצר את החיתוך הריק. אם בסוף התהליך החיתוך נשארת קבוצת חיתוך ריקה, המסקנה היא שהאלגוריתם אינו מסוגל למצוא את התרגום של המילה בשפת המקור. החיתוכים מתבצעים במקביל על המשפטים בשפת המקור ועל המשפטים המקבילים בשפת המטרה. נתבונן בדוגמא פשוטה מאוד:

הנתונים (שים לב שבעברית אותיות השימוש מופרדות מהמילה)

מילה	משפטים בשפת מקור	משפטים בשפת מטרה
1. ילדים	1. ה ילדים הלכו לבית הספר. 2. דיברנו עם ה ילדים אתמול. 3. ראיתי את ה ילדים בשבוע שעבר.	1. The children went to school. 2. We spoke with the children yesterday. 3. I saw the children last week.

חיתוך

חיתוך	מקור	מטרה
(1∩ 2)	ה ילדים * * *	The children * * *
((1∩ 2) ∩ 3)	ה ילדים * * *	The children * * *

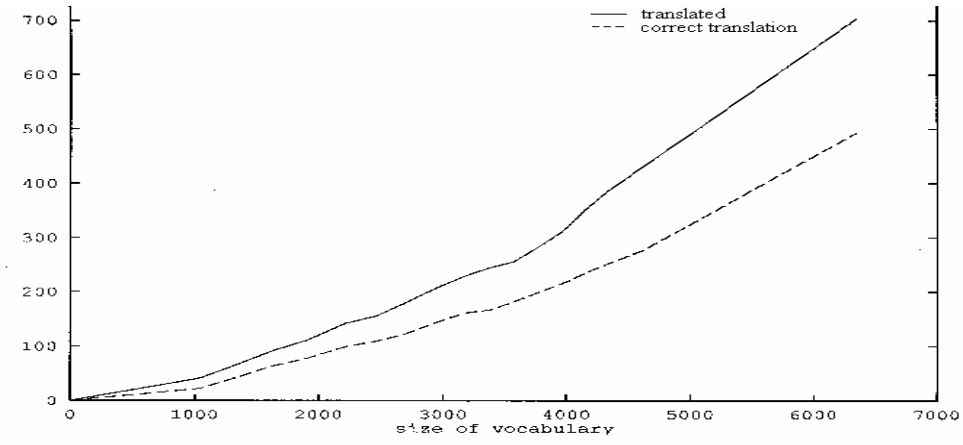
בסיום תהליך החיתוך, מכילה קבוצת החיתוך בשפת המקור את המילים *ה ילדים* ואילו קבוצת החיתוך בשפת המטרה מכילה את המילים *the children*, כלומר, **הביטוי** *ה ילדים* בעברית תורגם ל**ביטוי** *the children* באנגלית.

הדוגמא הנ"ל היא פשוטה וטריוויאלית, אולם בקורפוס האמיתי נתגלו ביטויים שלא היו מוכרים למתרגמים שאינם מומחים בגיאולוגיה ו/או גיאוגרפיה. דוגמא לביטוי כזה היא *זבל ירוק* שתורגם ל green manure. גילוי הביטויים והכללתם במילון שיפר מאוד את איכות התרגום של מערכת BITS.

הדוגמא מראה גם כי קבוצת החיתוך שומרת על **סדר** המילים במשפט. כתוצאה מכך ניתן לגלות גם ביטויים לא רציפים כגון: לא * ולא * (תרגום * *neither* * *nor*) כאשר הכוכבית מסמלת מילה כלשהי למשל: *לא כאן ולא שם* (*neither here nor there*).

בניגוד לניסוי של Brown et al. שנערך רק בכיוון אחד – תרגום מאנגלית לצרפתית בלבד – הופעל האלגוריתם המתואר למעלה בשני הכיוונים כלומר נערכו ניסיונות להשתמש בעברית כשפת מקור ובאנגלית כשפת מטרה וגם ניסיונות להשתמש באנגלית כשפת מקור ובעברית כשפת מטרה. לאחר שיוצרו מילון עברי-אנגלי ומילון אנגלי-עברי, נבנה מילון שלישי שכלל את הערכים המשותפים לשני המילונים. כ 80% מהערכים היו משותפים, ו 99.5% מהערכים המשותפים הכילו תרגומים נכונים.

מכיוון שהניסוי נערך על קורפוס קטן מאוד, המילון הסופי כיסה רק כ 15% מאוצר המילים. עם זאת ניסיונות על קורפוסים מצטברים מראים שככל אוצר המילים גדל כך גדלה כמות התרגומים ואיכותם. למעט בקורפוסים הקטנים ביותר, אחוז המילים המתורגמות נכון נשאר פחות או יותר קבוע (כ 70%). הציר האופקי של הגרף מייצג את גודל אוצר המילים, הציר האנכי מייצג את מספר המילים שתורגמו:



ניסוי שנערך על כ 10,000 כותרים של מאמרים מדעיים בגיאולוגיה (כ 75,000 מילים שוטפות), הראה כי אוצר המילים בכותרים החל להתקרב לרוויה לקראת הכותר ה 10,000. תוצאה זו מאפשרת להניח שהפעלת האלגוריתם על כ 20,000 כותרים תאפשר לייצר מילון שיקיף כ 70% מאוצר המילים בדיוק כמעט מוחלט שלא יצריך עריכה אנושית.

בדומה לשיטה של .Brown et al החיסרון העיקרי של האלגוריתם הנ"ל הוא הצורך בקורפוס דו-לשוני "גדול" המחולק למשפטים מקבילים. כיום, בסביבת ה Internet קל למצוא קורפוסים כאלה, ביניהם מסמכים של האיחוד האירופי ושל האו"ם הכתובים במספר שפות מקבילות.

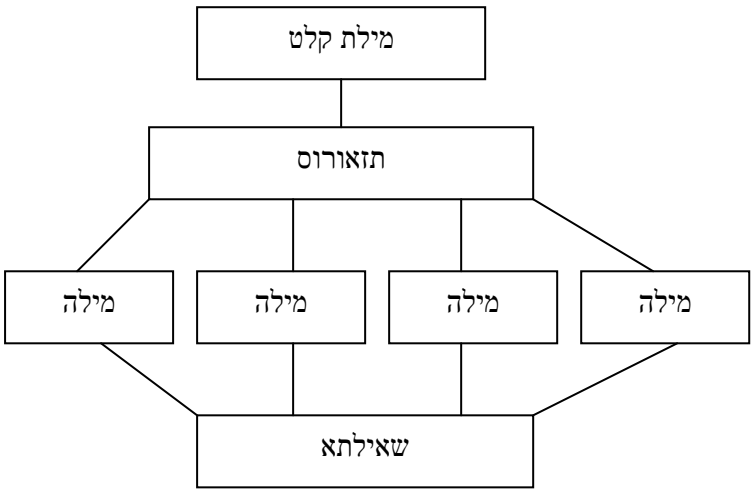
הערך של התוצאה הסופית, המילון, ברור מאליו. אולם, גם לתוצאות הביניים של תהליך ההפקה ערך משלהן. גילוי של ביטויים אופייניים לסוג מסויים של טקסט יכול לשפר תהליכים של סגמנטציה[2] ושל אחזור. רשימת הביטויים מראה כי ייתכן שכדאי להתייחס לסדרה מסויימת של מילים כמילה בודדת ולכלול באינדקס את הביטוי "זבל ירוק" במקום להתייחס לכל מילה בנפרד. לכל הפחות, ניתן לשמור את רשימת הביטויים בקובץ נפרד ולאפשר למשתמשים לבחור את הביטויים הנחוצים להם תוך כדי אחזור.

מבחינה טכנית, קל מאוד לתכנת את האלגוריתם (אורך התוכנית המקורית כ 300 שורות קוד ב Quick Basic) והביצועים מהירים ויעילים. בניית התוכנית מאפשרת להריץ את התהליך בצורה מחזורית ולהשתמש בתוצאות על מנת לעדכן ולשפר מילונים קיימים.

<u>תזאורי</u>

תזאורוס מייצג מודל ידע של תחום. מודל הידע מורכב ממושגים ומיחסים בין המושגים. מושג הוא ייצוג של רעיון המתקבל מהכללה של מקרים פרטיים. מושגים מבוטאים במילים או ביטויים. הסבר זה, המהווה למעשה הגדרה תיאורטית של תזאורוס, ממחיש מדוע קשה (ויקר) כל כך לבנות ולתחזק תזאורוס: על מנת לבנות מודל ידע של תחום דרוש ידע מקיף והכרה של המושגים המקצועיים באותו תחום. כלומר, רק מומחה או קבוצה של מומחים בתחום מסויים יכולה לבנות תזאורוס. שירותיהם של מומחים, שבדרך כלל בניית התזאורוס אינה עיסוקם העיקרי, הם מוגבלים ויקרים.

תזאורוס במערכת אחזור טקסט מתפקד כסוג של מילון: משתמשים מזינים למערכת מילות שאילתא מלוות בבקשה להרחבה באמצעות תזאורוס. המערכת מחפשת כל אחת ממילות הקלט בתזאורוס ובונה קבוצה של מילים שאותן יש לחפש בטקסט:



[2] בייחוד כאשר הסגמנטציה היא מבוססת מילון.

לכן, כאשר אנו ניצבים בפני המשימה של פיתוח תזאורוס למערכת אחזור טקסט, הבעיה העיקרית העומדת בפנינו היא, למעשה, סיווג של המילים בקורפוס, כך שחיפוש של מילה מסויימת בתזאורוס יחזיר קבוצה של מילים קרובות, ושימוש בכל אחת מהמילים בקבוצה זו בשאילתא (במקום במילת הקלט בלבד) ישפר את תוצאות האחזור. קבוצות המילים שאנו מחפשים צריכות להתאים לתחום הידע בו עוסקת מערכת האחזור ולא להיות כלליות מדי.

כיצד בונים קבוצות משמעותיות של מילים בצורה אוטומטית? במשך השנים נעשו ניסיונות רבים לסווג מילים בצורה כזאת או אחרת. מרבית הניסיונות הסתמכו על שכיחות יחסית והשתמשו ב cluster analysis על מנת לייצר "אשכולות" של מילים קרובות. השיטה המתוארת להלן, שונה משיטות אחרות לסיווג אוטומטי של מילים בכך שהיא מתבססת על ההגדרה הבלשנית של קטגוריה הקובעת כי שתי מילים שייכות לאותה קטגוריה אם יש להן אותה תפוצה (distribution).

ההגדרה הבלשנית של קטגוריה אינה מפרטת מהי "תפוצה". כל מנת לייצר קטגוריות המבוססות על תפוצה, יש צורך לבנות הגדרה פורמלית של המושג שאותה ניתן להזין למחשב. השלב הראשון בבניית הגדרה כזאת הוא הגדרת מושג ה**סביבה**: סביבה של מילה במשפט מבוטאת במונחים של מספר המילים המופיעות לפניה ואחריה באותו משפט:

ילדים *גדולים* אינם בוכים.

הסביבה של המילה גדולים היא ילדים * אינם בוכים.

התייחסות לסביבה מאפשרת להגדיר את התפוצה של מילה כרשימת כל הסביבות בהן מופיעה המילה בקורפוס. קבוצה (או קטגוריה) של מילים תורכב מכל המילים המופיעות באותה סביבה.

על מנת לזהות קבוצות של מילים דרוש קורפוס גדול[3] המחולק למשפטים. כמו כן יש להגדיר פרמטרים של סביבה: מספר המילים המופיעות לפני המילה שאותה מנסים לסווג ומספר המילים המופיעות אחריה (לדוגמא מילה אחת לפני ומילה אחת אחרי מהוות סביבה של 1~1). לאחר שנקבעו פרמטרים אלה ניתן "לאסוף" את כל המילים המופיעות באותה סביבה.

כתוצאה מתהליך האיסוף נוצרת רשימה של קבוצות מילים בעלות סביבה משותפת. בחינה של הקבוצות מראה כי הקבוצות שנוצרות הן בעלות משמעות סמנטית. בתזאורוס קנוי, המילים בכל קבוצה רשומות כ related terms. תהליך האיסוף הופעל על קורפוסים מסוגים שונים (ביבליוגרפיות בגיאולוגיה, מדריכים למשתמש של מערכת UNIX, התנ"ך, כתבות מעיתון הארץ...) והתופעה חוזרת על עצמה בכל אחד מהם. בקורפוס של כותרים של מאמרים מדעיים בגיאולוגיה נוצרו, לדוגמא, הקבוצות הבאות:

EOCENE,
JURASSIC,
MIOCENE,
PALEOZOIC,
TRIASSIC
********** (שמות של תקופות גיאולוגיות)

BASIN,
GULF,
PENINSULA,
SHIELD
********** (שמות של צורות שטח)

[3] דיון קצר בגודל הרצוי של הקורפוס מופיע בסעיף המתאר ייצור אוטומטי של מילונים דו-לשוניים.

מדריכים למשתמש של מערכת UNIX יצרו את הקבוצות הבאות:

LEFTMOST
RIGHTMOST
********** (כיוונים)

CATCH
IGNORE
********** (פעולות המבוצעות על סיגנלים)

בעברית נוצרו קבוצות שכללו, בין השאר, וריאציות שונות של כתיב:

בניה
בנייה
****

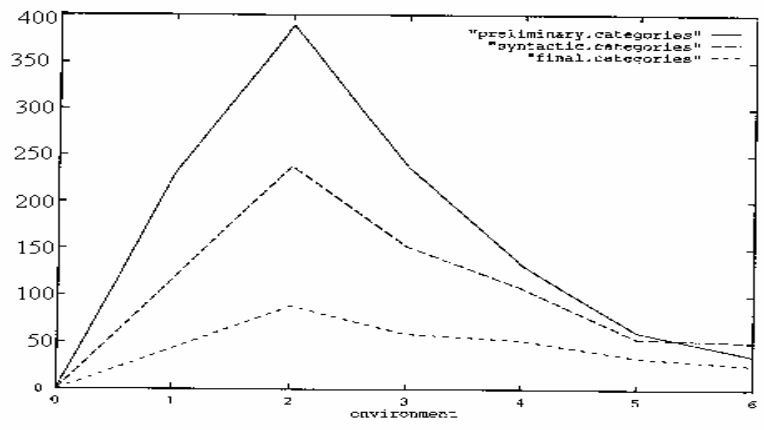
וגם קבוצות של הפכים כמו:

דרומי
צפוני
****

לא ניתן לזהות בצורה אוטומטית את סוג היחסים בין המילים בקבוצה מסויימת. לא ניתן להבחין האם המילים בקבוצה מסויימת הן תרדיפים, הפכים או מושגים ספציפיים המפרטים מושג על. למרות זאת, הניסיון מראה כי שימוש בקבוצות אלו, עוד לפני התערבות אנושית, עשוי לשפר את תוצאות האחזור של מערכת אחזור טקסט. במקרה הגרוע ביותר נוצרה לפחות תשתית לתזאורוס המתאים לדרישות המערכת והבלשן האחראי על תחזוקת המערכת קיבל "חומר גלם" שאותו ניתן לעבד.

שיטת הבנייה של קטגוריות סביבתיות מעוררת מספר בעיות:

הבעיה הראשונה היא שיש לקבוע מראש פרמטרים של סביבה. הגרף הבא מתאר מדידות למציאת הפרמטרים האופטימליים של הסביבה הציר האופקי מייצג את גודל הסביבה במונחים של מספר מילים (סביבה של 2 פירושה שתי מילים לפני, או מילה לפני ומילה אחרי או שתי מילים אחרי) הציר האנכי מייצג את מספר הקבוצות שנוצרו:



הגרף מראה כי מספר הקבוצות בסביבה של 2 הוא הגבוה ביותר והוא חריג. התופעה חזרה על עצמה בכל הקורפוסים שנבחנו. בחינה של הקבוצות שנוצרו בסביבה של 2 מראה כי הקבוצות אינן שונות בצורה משמעותית מקבוצות שנוצרו בסביבות אחרות ולכן לא ניתן לומר כי סביבה של 2 היא הסביבה האופטימלית עם זאת סביבה של 1~1 היא "האינטואיטיבית" ביותר.

הבעיה השנייה בשיטה נוגעת למילים המופיעות ביותר מקבוצה אחת. כאן מתעוררת השאלה כיצד יש לסווג מילים כאלה? האם יש לאחד קבוצות המכילות את אותה מילה, או אולי יש להתייחס רק לקבוצות מילים החוזרות על עצמן? לשאלות אלו אין תשובות מוחלטות. ההחלטות הסופיות תלויות בתוצאה הרצויה: אם ברצוננו להגדיל את ה recall של מערכת אחזור טקסט ניתן "לשרשר" (chain) קבוצות מילים המכילות לפחות מילה אחת משותפת. אם, לעומת זאת, ברצוננו להגדיל את ה precision של מערכת האחזור, ניתן להשתמש רק בקבוצות החוזרות על עצמן גם אם הן תת-קבוצות של קבוצות גדולות יותר.

החיסרון העיקרי של שיטת הסיווג הנ"ל היא הצורך בקורפוס גדול של טקסט המחולק למשפטים, אולם כמו שכבר הוזכר, היום בתקופת האינטרנט קל מאוד לבנות קורפוס כזה, בייחוד אם הטקסטים שבו הם בשפה אחת בלבד.

יתרונותיה של שיטת הסיווג הם שהיא פשוטה, זולה וקלה לביצוע. מיסוד תהליך הסיווג מאפשר לחזור ולהשתמש בו בצורה מחזורית על מנת לעדכן תזאורי קיימים ולשפר את תוצאות האחזור. יתרון נוסף הוא שתוצאות הסיווג מספקות נקודת מבט חדשה ושונה על הקורפוס. באמצעות התוצאות ניתן לעיתים לזהות מילים שבשפה יומיומית אינן נחשבות לקרובות, אך בתחום ידע מסויים הן יכולות לשמש (לדוגמא) הפכים:

מדריכים למשתמש של מערכת UNIX יצרו את הקבוצה הבאה:

CATCH
IGNORE
****

בשפה יומיומית, לא ניתן להבחין בקרבה סמנטית בין המילים *לתפוס* ו-*להתעלם*. בחינה של הסביבה שיצרה את הקבוצה: *signals currently being * or* מראה שבשפה המקצועית של המדריכים למשתמש של מערכת UNIX שתי המילים הן הפכים כלומר בעולם של מערכת UNIX ניתן *לתפוס* אותות או להיפך *להתעלם* מהם.

סיכום

המאמר מציג שתי שיטות פשוטות ליצירת כלים בלשניים, היכולים לסייע באחזור טקסט, מקורפוס גדול. שיטות אלו הן רק חלק ממערכת גדולה יותר לייצור ידע בלשני הנקראת FROG. מערכת FROG משתמשת בשיטות דומות של קיבוץ אלמנטים, חיתוך קבוצות וספירת שכיחויות על מנת להפיק רשימות של קידומות, סיומות, כללי איות וכו'. השיטות שבהן משתמשת מערכת FROG יושמו במקומות שונים והוכחו כיעילות. נזכיר כאן רק שכללי המורפולוגיה וכללי הסאונדקס של מנוע האחזור של FLAIR™ פותחו בשיטות שבהן משתמשת מערכת FROG.

יתרונותיהן של השיטות שתוארו הן גם חסרונותיהן. הרעיונות העומדים מאחורי השיטות פשוטים מאוד ומספקים תוצאות חשובות ומעניינות, אולם, דווקא בגלל פשטותן של שיטות העיבוד, יש להתייחס לתוצאות המתקבלות בזהירות רבה ולהקפיד על ביקורת אנושית מדוקדקת לפני השימוש בהן. שיטות העיבוד מותירות מספר בעיות פתוחות כגון הגדרת גודל הקורפוס האידאלי למערכת מסויימת וקביעת הפרמטרים האופטימליים של סביבת הסיווג. נותר גם שדה נרחב למחקר ולפיתוחים נוספים ביניהם:

- שיפור תהליך יצירת המילון כך שניתן יהיה לזהות הומוגראפים ומילים שלהן יותר מתרגום אחד.
- מציאת שיטה אוטומטית להגדרת היחסים בין מילים בקבוצה מסויימת על סמך הסביבה.
- יצירת רשתות סמנטיות של מושגים על-ידי שירשור של קבוצות מילים.

ביבליוגרפיה

Arad, I. (1991). BITS, A Hebrew-English Bibliographic Translation System. *Machine Translation*, 6 pp. 247-263.

Arad I. (1991). A Quasi-Statistical Approach to Automatic Generation of Linguistic Knowledge. Ph.D Thesis, UMIST, UK.

Brown, P., Cocke, J., Della Pietra, S., Della Pietra, V., Jelinek, F., Mercer, R., Roossin P. (1988). A statistical approach to language translation. In*:* *Proceedings of COLING Budapest*, 1988, pp. 71-76.

Tsujii, J.I., Ananuadou, S., Arad, I., Sekine, S. (1992). Linguistic Knowledge Acquisition from Corpora. In: *Proceedings of the International Workshop on Fundamental Research for Future Generation of Natural Language Processing FGNLP*, Manchester, 1992, pp. 61-81.

Zipf, G.K. (1949). Human behaviour and the principle of least effort. Cambridge Mass, Addison-Wesley.

נספח – הגדרות

מונח עברי	מונח אנגלי	הגדרה/הסבר
דקדוק	Grammar	קבוצה של כללים המשמשת לתרגום מילה או משפט מייצוג אחד לייצוג אחר.
יחס	Relation	נקודת מבט או תכונה הקושרת בין שני מושגים בתזאורוס.
מושג	Concept	ייצוג של רעיון המתקבל מהכללה של מקרים פרטיים.
מילה	Word	סימן מוסכם. סדרת אותיות בעלת משמעות.
מפתח בלשני של מילה	Key	ייצוג של מילה המשותף לכל הוריאציות של אותה מילה. כל סוג וריאציות (וריאציות כתיב, וריאציות מורפולוגיות, וריאציות פונטיות) מיוצג על ידי מפתח משלו.
ערך בתזאורוס	Thesaurus entry	אוסף של שדות המכיל את כל המידע הקיים על מושג מסויים.
רשימת הרחבה של מילה	Word expansion set	קבוצה של מילים בעלות מפתח בלשני משותף.
תזאורוס	Thesaurus	ייצוג של מודל ידע של תחום מסויים. מודל הידע מורכב ממושגים ומיחסים בין המושגים.

איתור נושא מסמך בצורה אוטומטית תוך שימוש במילים נפוצות

עפר דרורי

שימוש במילים נפוצות במסמך לאיתור נושא המסמך

עפר דרורי
שע"ם / האוניברסיטה העברית בירושלים
ת.ד. 10414 ירושלים 91103

עולם מאגרי המידע מכיל כיום מאות מיליונים של מסמכים אלקטרוניים. קטלוג איכותי של מידע הוא ידני והוא כרוך במשאבים רבים הגורמים לכך שרוב המידע כיום אינו מקוטלג. איתור מידע ברשת האינטרנט היום בעייתי בעיקר בגלל ריבוי המסמכים בה. ממחקר אחר נמצא כי שיוך מסמכים לנושא יכול לשפר את יכולת איתור המידע ברשת. מטרת המאמר להציג כלי לאיתור נושא מסמך בצורה אוטומטית ולהציג תוצאות בדיקה שנערכה על בסיס תוכנה שפותחה לנושא זה.
מילות מפתח: קטגוריזציה של טקסט, מילים נפוצות, איתור אוטומטי של נושא מסמך

1. מבוא

כמות דפי המידע ברשת האינטרנט עפ"י הערכת מומחים היא למעלה ממיליארד וזאת מבלי לקחת בחשבון את המידע בבסיסי נתונים הנמצאים ברשת. בהיקף כזה של מידע קשה לצפות שאפילו אחוז ניכר מהמאגר יהיה מקוטלג בגלל המשאבים הרבים הנדרשים לקטלוג מידע. בעולם בו כמות המידע הולכת וגדלה בקצב מסחרר קטלוג דפי מידע ומסמכים בצורה אוטומטית מתבקש וחיוני במיוחד לצורך איתור מידע.

קטלוג מידע הוא פעולת שיוך מידע לרשימת נושאים או מושגים קבועה מראש. חלוצי הקטלוג של הידע האנושי היו אנשי הספריות אשר פתחו שיטות שונות לשיוך אוצר הספרים (ובהמשך גם מדיות אחרות) עפ"י נושאים. בין שיטות הקטלוג הנפוצות והוותיקות: Dewey (המחלקת את עולם הנושאים עפ"י קבוצות מספרים עשרוניים) ו- LC (שיטת הקטלוג של ספריית הקונגרס האמריקאי המבוססת על רשימת מונחים). לצד שיטות מסורתיות אילו התפתחו כלים נוספים שנועדו לשייך טקסטים קיימים לקטגוריות נתונות. אחד הכלים הנפוצים יותר הוא מדריך האתרים Yahoo המבצע פעולת קטלוג ידנית על דפי מידע רבים ברשת. בגלל פעולת הקטלוג הידנית הכיסוי של מנוע כזה הוא מוגבל ביותר.

מספר מחקרים נעשו בתחום ארגון המידע בצורה אוטומטית ורובם מתייחסים לרשת האינטרנט והצגת מידע ממנה. Allen פיתח אב טיפוס להצגת תוצאות חיפוש העושה שימוש בשיטת Dewey (Allen, 1995). בפרויקט SuperBook אורגנו פסקאות טקסט בטבלת תכולה היררכית הדומה לתוכן עניינים (Landauer et al. 1993). יצירת תוכן עניינים דומה

לחיפושים בספריית הקונגרס נעשתה ע"י (Marchionini et al. 1998). במערכת WebCutter מוצגת מפת החיפוש של המשתמש על בסיס קטגוריות שונות (Maarek et al. 1997).
דרך אחרת לארגון קבוצת מסמכים בצורה אוטומטית היא באמצעות Clustering. בפעולה זו מאורגנים קבוצות מסמכים על בסיס דמיון ביניהם ולא על בסיס רשימת קטגוריות קבועה מראש. קיימים פרויקטים רבים בתחום אך מכיוון שהם נוגעים פחות לפעולת הקטלוג שבה עוסק המאמר אציין כמה מראי מקום בלבד (Zamir & Etzioni 1998), (Zamir & Etzioni 1999), (Hearst & Pedersen 1996).
דרך שלישית לארגון קבוצות מסמכים נקראת Classification. בשיטה זו נעשה שימוש בטכניקות סטטיסטיות הפועלות על מסמכים שנקבעה להם קטגוריה באמצעים אחרים. המערכת לומדת את התנהגות המסמכים ביחס לקטגוריה שנקבעה ומאפשרת לייצר קטגוריה דומה על מסמכים שלא קוטלגו מראש. גם שיטה זו נוגעת פחות לתחום המאמר וגם כאן יצוינו מספר מקורות בלבד (Chekuri et al. 1997), (Mladenic 1998), (Chen & Dumais 2000).

2. איתור נושא מאמר

במסגרת סדרת ניסויים שנועדו לבדוק את הדרך האפקטיבית ביותר להצגת מידע ברשימת תוצאות חיפוש נמצא כי הצגת נושא המסמך או הקטגוריה אליה הוא משתייך יכולה להביא מספר תועלות למשתמש (Drori, 2000). היתרון העיקרי הוא שבעת העיון ברשימה יכול המשתמש לאתר את המסמך המעניין אותו בהקשר לשאלת החיפוש וזאת מבלי צורך לקרוא את כלל המסמכים ברשימה. איתור נושא המסמך לשם הצגתו אפשרי במספר דרכים כאשר הדרך המקובלת ביותר מתבססת על אפיון ידני של המסמך לפי קטגוריות שונות. במסמכים מדעיים מחבר המסמך מציין את מילות המפתח שלו, במאגרים מדעיים מוסיף צוות המאגר את רשימת המונחים הרלוונטיים למאמר (ראה לדוגמא Index Terms ב- ACM Digital Library) ובמדריכים ממוחשבים באינטרנט מוסיף צוות המדריך את הקטגוריה אליה משתייך המסמך (ראה לדוגמא Yahoo). לצד השיטות הידניות המדויקות יחסית אך הגוזלות משאבים רבים, קיים צורך באפיון ממוחשב אשר יאפשר קטלוג מסה של מסמכים ממקורות שונים כולל כמובן כאלו שאינם ממופתחים או מקוטלגים.

לצורך הניסויים שנערכו פותח כלי ממוחשב המבצע ניתוח של טקסט המסמכים הנדרשים בצורה אוטומטית לחלוטין כאשר התוצאה שלו כוללת את המילים המשמעותיות במסמך אשר מרכיבות בפועל את נושא המסמך. התוכנה מבצעת "קריאה" של טקסט המסמך וקובעת על בסיס ניתוח סטטיסטי מהם המילים המשמעותיות ביותר במסמך. ניתוח הטקסט נעשה לאחר השמטת מילים חסרות ערך לנושא תוך שימוש ב- Stop List וכן טיפול מוגבל במרכיבים לשוניים. התוכנה יודעת לטפל במסמכים בשפה האנגלית והעברית וכן במסמכים דו לשוניים בשתי השפות. התוכנה יודעת לטפל במרכיב הסטטיסטי שלה בכל שפה ובמידת הצלחה פחותה בכל הקשור למרכיב הלשוני. הטיפול הלשוני המוגבל בא לידי ביטוי בהכרת צורת הטיות של השפה ובמילים שאינן משמעותיות שיש לנפות באמצעות

Stop list הייחודי לכל שפה.

3. מחקר

כדי לוודא עד כמה ניתן להתבסס על ניתוח המסמכים של התוכנה בקביעת נושא המסמך נבחרו שני אוספים של מסמכים מדעיים בהם ניתן לקבל בצורה ישירה את נושא המסמך תוך התבססות על מילות המפתח של המסמך וכן על הכותרת שלו. השוואה בין נושא המסמך כפי שקבעה התוכנה לבין נושא המסמך כפי שהוגדר ע"י מחבר המסמך תגדיר את מידת ההצלחה של התוכנה. הבדיקה כללה ניתוח Full text של 100 מסמכים מדעיים משני תחומי מחקר: 1. General Management 2. Industrial Management. רשימת כתבי העת מהם נלקחו המאמרים ראה בנספח 1.

נערכה השוואה בין המילים המשמעותיות שאיתרה התוכנה מול שתי קבוצות מילים קיימות. קבוצה אחת כללה את מילות המפתח שקבע מחבר המאמר והקבוצה השנייה כללה את המילים המופיעות בכותרת המסמך (שגם היא נועדה לאפיין את המסמך בצורה ברורה). הצלחה בניבוי נושא של מסמך תחשב כאשר תתקיים התאמה גבוהה בין המונחים שהציעה התוכנה בהשוואה למונחים הקיימים במאמר (מילות מפתח או מילים בכותרת המסמך). לצורך אימות המידע נדרשו 2 אנשים מיומנים בתחום המאמרים לקרוא אותם ולהעריך על בסיס המילים המשמעותיות שבחרה התוכנה את מידת היכולת שלהם לקבוע את נושא המסמך מבלי לקרוא אותו.

המילים שדורגו ע"י התוכנה כמשמעותיות היו מילים שהופיעו בכל מאמר מעל 20 פעמים (במאמרים שאורכם הממוצע היה כ- 6000 מילים). לצורך הבדיקה חולקה רשימת הדירוג ל- 3 קבוצות של מילים: 3 המילים בראש הרשימה, 5 המילים בראש הרשימה (רשימת ה- 3 ועוד 2 מילים) ו- 10 המילים בראש הרשימה.

4. ממצאים

ניתוח הנתונים נעשה בנפרד לגבי כל תחום מדעי מתוך כוונה לבחון האם יש שוני בהתנהגות של טקסטים מתחומים שונים (ראה טבלה 1). לגבי תחום הניהול הכללי נמצא כי 50.4% מתוך מילות המפתח של כלל המאמרים זוהו ע"י התוכנה מתוך רשימת 10 המילים העדיפות ע"י התוכנה. כמו כן נמצא כי 49.7% ממילות כותרות המאמרים זוהו ע"י התוכנה מתוך רשימת 10 המילים העדיפות ע"י התוכנה. כאשר חוברו מילות המפתח והכותרות ונופו הכפולים אחוז הזיהוי הגיע ל- 43.9%. הממצאים לגבי 5 המילים המועדפות היו 56.4% זיהוי ולגבי 3 המילים העדיפות היו 70.6%. קיים הבדל מובהק בין הזיהוי של 10 מילים מול 5 מילים ובין 10 מילים ל- 3 המילים הנפוצות ($P < 0.0001$). קיים הבדל מובהק בין הזיהוי של 5 ו- 3 המילים הנפוצות ($P < 0.0015$).

	אחוז הזיהוי ממילות מפתח (SD)	אחוז הזיהוי ממילות כותרת	אחוז הזיהוי המשולב (מילות מפתח וכותרת)	אחוז הזיהוי המשולב ביחס ל-10 העדיפות	אחוז הזיהוי המשולב ביחס ל-5 העדיפות	אחוז הזיהוי המשולב ביחס ל-3 העדיפות
ניהול כללי	50.4 (0.27)	49.7 (0.21)	43.9 (0.18)	45.4 (0.18)	56.4 (0.22)	70.6 (0.28)
ניהול תעשייתי	46.0 (0.23)	46.7 (0.21)	40.8 (0.18)	44.2 (0.15)	55.9 (0.22)	64.1 (0.26)
כולל	48.3 (0.25)	48.3 (0.21)	42.3 (0.18)	44.8 (0.16)	56.1 (0.22)	67.4 (0.27)

טבלה מספר 1 - ריכוז אחוזי הזיהוי של התוכנה ביחס שבין טקסט המאמר לטקסט מילות המפתח וכותרות המאמרים

מכיוון שמטרת התוכנה היתה כאמור לאתר את נושא המסמך בצורה אוטומטית ע״י ניתוח הטקסט נערך אימות של קביעת התוכנה ע״י שני אנשים הבקיאים בתחומי המדע שנבדקו. המשתתפים התבקשו לחוות את דעתם האם ניתן להבין את נושא המאמר מבלי לקרוא אותו על בסיס המלצת התוכנה. ניתן לראות מטבלה 2 כי ניתן לזהות את נושא המסמך בלמעלה מ- 70% כאשר נתונות 5 מילים מומלצות. אחוז הזיהוי יורד כאשר מנסים לבצע את הזיהוי על בסיס 3 מילים בלבד. קיים הבדל מובהק בין זיהוי עפ״י 5 מילים ל- 3 מילים ($P < 0.0003$).

	אחוז זיהוי הנושא ע״י אדם ביחס ל-5 המילים העדיפות	אחוז זיהוי הנושא ע״י אדם ביחס ל-3 המילים העדיפות
ניהול כללי	68.6	52.9
ניהול תעשייתי	74.0	40.0
כולל	71.3	46.5

טבלה 2 - ריכוז אחוזי הזיהוי של נושא המאמר ביחס למילים העדיפות של התוכנה בהתפלגות של 3 ו- 5 מילים

המחקר בדק גם את הקשר בין גודל המסמכים ליכולת הזיהוי. בתחום הניהול הכללי המאמר הקצר ביותר היה בן 1537 מילים והארוך ביותר בן 16001 מילים. מספר המילים הממוצע למאמר בתחום היה 6324 מילים. ניתן לראות מנתוני טבלה 3 כי כאשר מספר המילים בטקסט גדול יותר יש עליה באחוזי הזיהוי של מילות המפתח, של המילים

המשולבות (מפתח וכותרת), של 5 ו- 3 המילים העדיפות שמציעה התוכנה . כמו כן ניתן לראות כי קיימת ירידה באחוזי הזיהוי של מילות הכותרת ו- 10 המילים המומלצות ע"י התוכנה. ניתן להסביר את הירידה בזיהוי במילות הכותרת בגלל כותרות לא אינפורמטיביות מספיק (לדוגמא נמצאה כותרות למאמר: "Can elephants fly ?") וכן בגלל הפיזור הגדול יחסית של 10 המילים העדיפות.

	אחוז זיהוי ממילות מפתח	אחוז זיהוי ממילות כותרת	אחוז זיהוי משולב מכותרת ומילות מפתח	אחוז זיהוי מ- 10 מילים עדיפות	אחוז זיהוי מ- 5 מילים עדיפות	אחוז זיהוי מ- 3 מילים עדיפות
>6324	44.7	50.9	40.9	48	55.4	70.7
<6324	57.4	48.3	41.4	42.3	57.7	71

טבלה 3 - ריכוז אחוזי הזיהוי ביחס לגודל המאמר (ניהול כללי)

בתחום הניהול התעשייתי המאמר הקצר ביותר היה בן 2294 מילים והארוך ביותר בן 9734 מילים. מספר המילים הממוצע למאמר בתחום היה 4915 מילים. ניתן לראות מנתוני טבלה 4 כי כאשר מספר המילים בטקסט גדול יותר יש עליה באחוזי הזיהוי של מילות המפתח, של מילות הכותרת, ושל 5 ו- 3 המילים העדיפות שמציעה התוכנה. כמו כן ניתן לראות כי קיימת ירידה באחוזי הזיהוי של 10 המילים המומלצות ע"י התוכנה. ניתן להסביר את הירידה בזיהוי של 10 המילים העדיפות בגלל הפיזור הגדול יחסית של 10 המילים העדיפות.

	אחוז זיהוי ממילות מפתח	אחוז זיהוי ממילות כותרת	אחוז זיהוי משולב מכותרת ומילות מפתח	אחוז זיהוי מ- 10 מילים עדיפות	אחוז זיהוי מ- 5 מילים עדיפות	אחוז זיהוי מ- 3 מילים עדיפות
>4915	45.2	45.5	48.8	44.7	55.5	60.7
<4915	47.1	48.3	38.7	43.6	56.4	68.2

טבלה 4 - ריכוז אחוזי הזיהוי ביחס לגודל המאמר (ניהול תעשייתי)

5. מסקנות

ממצאי המחקר מאפשרים להגיע למספר מסקנות

1. ניתן לזהות נושא מסמך בצורה אוטומטית באמצעות התוכנה על בסיס המלצה של התוכנה עד כדי 72% מהמקרים כאשר עושים שימוש בחמשת המילים המדורגות עליונות.
2. ככל שמספר המילים במסמך גדול יותר אחוזי הזיהוי של מילות המפתח גדל.
3. כאשר מחברים את מילות המפתח והמילים בכותרת קטן אחוז הזיהוי לעומת השימוש במילות המפתח בלבד או במילים של הכותרת בלבד.
4. יש עדיפות מעטה בזיהוי למילות המפתח על פני מילות הכותרת. מהבחינה היישומית קיימים מסמכים רבים שאין בהם רישום של מילות מפתח בעוד שרוב המסמכים כוללים כותרת. בגלל ההבדל הלא מובהק באחוזי הזיהוי בין מילות המפתל למילות הכותרת ניתן להשתמש במילות הכותרת כאשר מילות המפתח לא קיימות.
5. קיים שוני באחוזי הזיהוי בהתאם לסוג החומר. במקרה שלנו ברוב הנושאים שנבדקו אחוזי הזיהוי בתחום הניהול התעשייתי היו נמוכים מאחוזי הזיהוי מתחום הניהול הכללי. אם זאת יש לציין שאורך המסמכים הממוצע בתחום הניהול התעשייתי היה קטן יותר ב- 22.3%.
6. ניתן לאפיין בצורה אוטומטית נושא של מסמך באחוזי הצלחה של כ- 72%.
7. מומלץ למנהלי מאגרי מידע שאינם מכילים קטגוריות לבצע פעולת איתור קטגוריה בכלים אוטומטיים (כדוגמת התוכנה שבמחקר) ורישומה לתועלת המשתמשים במנועי החיפוש השונים

6. סיכום והמשך מחקר

מטרת המחקר היתה לבדוק האם ניתן בכלים אוטומטיים לזהות נושא של מסמך. לצורך המחקר פותחה תוכנה (TextAnalysis) המבצעת ניתוח סטטיסטי של מילים בטקסט נתון ומייצרת רשימת מילים משמעותיות האמורות לייצג את נושא המסמך. נבדקו 100 מאמרים מדעיים בשני תחומי מחקר וחושבו אחוזי הזיהוי של התוכנה מול מילות המפתח והמילים בכותרת. אורך המסמך הוא אחד הפרמטרים שיכולים להשפיע על אחוזי הזיהוי.
המחקר הנוכחי מציב אתגר למפתחי התוכנה להגדיל את אחוזי הזיהוי. נרשמו מספר הערות לשיפור התוכנה הכוללות התייחסות לטיפול בביטויים, ראשי תיבות, מילים חסרות ערך ששוקללו ועוד. כמו כן יעשה ניסיון לשיפור האלגוריתם לצורך שיפור בביצועי התוכנה.
נושא למחקר עתידי הוא הגדרת נושא המסמך ב 3-2 מילים בלבד מתוך החמש המשמעותיות תוך התייחסות למונחים מקצועיים המקובלים בתחום.

תודות

ברצוני להודות לישראל מבשב על עזרתו בפיתוח התוכנה ולשירלי שרביט ורחל מופי על עזרתן בביצוע המחקר.

הערות

ACM Digital library - http://www.acm.org/dl/

Allen, R., Two digital library interfaces that exploit hierarchical structure, *Proceedings of DAGS95: Elctronic Publishing and the Information Superhighway*, 1995.

Chekuri, C. et al., Web search using automated classification, *Sixth International World Wide Web Conference*, Santa-Clara, CA, 1997.

Chen, H., Dumais, S., Bringing order to the web: automatically categorizing search results, *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'00)*, 2000.

Drori, Offer. "Using Text Elements by Context to Display Search Results in Information Retrieval Systems", *Information Doors - Where Information Search and Hypertext Link (a workshop proceedings held in conjunction with the ACM Hypertext 2000 and ACM Digital Libraries 2000 conferences)*, May 2000, San Antonio, Texas, USA, 17-22.

Hearst, M., Karadi, C., Searching and browsing text collections with large category hierarchies, *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'97)*, Atlanta GA, 1997.

Maarek, Y., et al., WecCutter: a system for dynamic and tailorable site mapping, *Proceedings of the 6th International World Wide Web Conference*, Santa-Clara CA, 1997.

Marchionini, G., et al., Interfaces and tools for the Library of Congress national digital library program, *Information Processing and Management*, 34, 1998, 535-555.

Mladenic, D., Turning Yahoo into an automatic web page classifier, *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI'98)*, 1998, 473-474.

Landauer, T. et. al., Enhancing the usability of text through computer delivery and formative evaluation: the SuperBook project, *Hypertext - A Psychological Perspective*, Ellis Horwood, 1993.

Yahho - http://www.yahoo.com

Zamir, O., Etzioni, O., Grouper: A dynamic clustering interface to web search results, *Proceedings of WWW8*, Toronto, Canada, 1999.

Zamir, O., Etzionin, O., Web document clustering: a feasibility demonstration, *Proceedings of the 19th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR98)*, 1998, 46-54.

נספח 1 - רשימת כתבי העת שהשתתפו במחקר

General management

European Journal of Innovation Management
Management Decision
European Business Review
Accounting Auditing & Accountability Journal
Disaster Prevention and Management
International Journal of Entrepreneurial Behavior & Research
Environmental Management and Health
Journal of Management History

Industrial management

Industrial Management & Data Systems
Integrated Manufacturing Systems
Logistics Information Management
International Journal of Operations & Production Management
International Journal of Agile Management Systems
International Journal of Service Industry Management
International Journal of Manpower

עידון תהליכי חיפוש במאגרי מידע והצגתם למשתמש

זיו סלייטר ורחל בן עזרא

מכללה אקדמית הדסה
החוג למדעי המחשב

פרוייקט גמר

עידון תהליכי חיפוש במאגרי מידע והצגתם למשתמש

מוגש על ידי: זיו סלייטר

רחל בן-עזרא

מנחה: ד''ר עופר דרורי

ירושלים, תשרי תשס''ד, אוקטובר 2003

תודות

תודה לד"ר עופר דרורי, על כך שהנחה אותנו במהלך הפרוייקט.

תוכן עניינים

רשימת נספחים

1. מבוא

המידע הנמצא באינטרנט זמין ומוצע לכל. במהלך השנים, כמות המידע הנגיש גדלה וכך גם מספר הגולשים. גולשים המעונינים לאתר כיום מידע בנושא ספציפי, יעשו זאת באמצעות הצגת שאילתא רלוונטית באחד ממנועי חיפוש הקיימים בשוק, בציפייה שאלו יובילו אותם אל האתרים/דפים העונים לבקשה שהגדירו. התוצאות המתקבלות ממנועי חיפוש אלו, ע"פ רוב, אינן מושלמות. לעיתים קרובות, מנועי החיפוש נותנים בידינו תוצאות מהן נעדרים אתרים/דפים הרלוונטיים לשאילתא שהוצגה, או לחילופין כוללות אתרים/דפים אשר אינם עונים על בקשת המשתמש.

ככל שעובר הזמן, היקף המידע הקיים ברשת גדל, ואתו מתעצם ומתחדד הצורך בסידור וארגון התוצאות המתקבלות ממנועי החיפוש. אי לכך, בחרנו בתכנון מערכת המארגנת וממיינת את המידע המתקבל בתוצאות החיפוש שביצעו המנועים הקיימים, לקטגוריות נושאיות.

1.1. הבעיה הכוללת והבעיה הפרטית

1.1.1. הבעיה הכוללת

האינטרנט הנו מאגר מידע עצום בגודלו, אשר נמצא, כל העת, במגמת צמיחה והתרחבות. מערכות לשליפת נתונים ספציפיים, מתוך מאגר מידע נרחב זה, הנן פופולריות במיוחד, היות והן מהוות את הכלי האולטימטיבי לאיתור המידע הרלוונטי עבור המשתמש. מערכות אלו מתוכננות ומעוצבות במטרה להקל על שליפה של מידע עבור מגוון המשתמשים ברשת.

קיומו של פער תפיסתי טבעי בין כותבי מאמרים המופיעים ברשת ובין משתמשי הרשת, מוביל למצב בו תוצאות שליפה שגרתית של הדפים המבוקשים ממאגר המידע, אינן עונות על צרכי משתמשי הרשת, או עונות באופן חלקי בלבד.

ביטוי לפער תפיסתי זה, נמצא בשימוש שעושים כותבי המאמרים ומשתמשי הרשת, בשפה. כותבי מאמרים ומשתמשי-הרשת, משתמשים תכופות במילים שונות בעלות אותה משמעות או מציגים משמעויות שונות לאותה מילה. כתוצאה מכך התוצאות המתקבלות במערכת שליפת נתונים, מכילות תשובות שגויות. כך לדוגמא, המילה "apple" מכילה שתי משמעויות שונות - "apple computer" וכן משמעות שונה לגמרי "apple pie". שימוש במילה "apple" במנועי חיפוש יוביל לשליפת כמות גדולה של דפים, אשר רק מיעוטם ייתן מענה לצורכי המשתמש, ולסינונם של מסמכים רלוונטיים רבים עקב "חוסר התאמה" שהוגדרה במערכת (התאמה שלילית). כן, הביטוי "airline schedule" לא יתקבל בתוצאות לשאילתא בה הוגדר הביטוי "airplane schedule" , בשל "חוסר ההתאמה" שהוגדר במערכת, למרות שלשני הביטויים משמעות סמנטית זהה.[1]

אם כן, כאשר משתמש עורך חיפוש ברשת באמצעות אחד ממנועי החיפוש הקיימים הוא צפוי לקבל מספר רב של תשובות לשאלה נתונה אחת. מנוע החיפוש (המכיל מאגר מידע משלו) יציג לו רשימה ארוכה של תוצאות בצורה סדרתית, כאשר לרוב לא תצוין עדיפות לתוצאה אחת על פני תוצאה אחרת. רשימה זו

כוללת, בד"כ, פרטים מועטים על כל מסמך (לרוב כותרת, כתובת המסמך, ומספר שורות המכילות את ערך החיפוש) כך שבסקירה ראשונית של הרשימה אין למשתמש אפשרות להחליט אילו רשומות רלוונטיות לגביו ואילו לאו. היות ואין בנמצא מערכת המסננת בצורה מושלמת את התוצאות הרלוונטיות בלבד עבור המשתמש, ובהעדר קטגוריזציה של תוצאות החיפוש המתקבלות, רוב המשתמשים במנועי החיפוש בודקים את 20-10 התוצאות הראשונות המתקבלות ברשימה, תוצאות אלו לא בהכרח יכילו את המידע המבוקש. כתוצאה מכך הזמן הנדרש לקבלת המענה הרלוונטי למשתמש מתארך, החיפוש מסתרבל ולעיתים יפסיק המשתמש את החיפוש – ויצא בידיים ריקות. [2]

1.1.2. הבעיה הפרטית

על מנת לאפשר למשתמש בכל זאת להגיע אל מבוקשו ולמקד אותו אל המסמכים הרלוונטיים עבורו, דרוש ממשק אשר יציג בצורה יעילה יותר את תוצאות החיפוש, ויסנן מסמכים לא רלוונטיים. מממשק זה נדרש להיות פשוט לתפעול בכדי לאפשר גם לאנשים שאינם מיומנים בחיפוש להגיע אל מבוקשם. בניית ממשק שכזה דורשת איתור מסמכים אשר מלבד התאמתם למילות החיפוש, הם גם הרלוונטיים ביותר עבור המשתמש.

ניתן יהיה לבצע איתור שכזה אם כל מסמך במאגר הנתונים הרחב, יקוטלג תחת שם נושא מסוים/ קטגוריה, הלקוח/ה מתוך קבוצת קטגוריות מוגדרת מראש. הקטגוריות שבהן עוסקים המסמכים העונים למילות החיפוש שהגדיר המשתמש, יוצגו לו. לאחר מכן, יוכל המשתמש לבחור לצפות רק במסמכים העוסקים בנושא/ים הרלוונטי/ים עבורו. בכך למעשה, תאפשר המערכת למשתמש להתמקד רק במסמכים העונים לצרכיו, ולהתעלם ממסמכים רבים אשר ייתכן שעונים לתנאי החיפוש, אך אינם מעניינים אותו.

ישנם מנועי חיפוש דוגמת Yahoo!, אשר בסיועם של אנשי מקצוע מבצעים סיווג ידני של חלק מהמסמכים הקיימים במאגר שלהם, תחת קטגוריות אשר הוגדרו על ידיהם. בכך מאפשר Yahoo! למשתמש לקבל מידע נוסף על מסמכים אלה, או לחילופין לחפש מסמכים ע"פ הקטגוריה. עם זאת, ביצוע הקטלוג בצורה ידנית הנו מוגבל ואינו מאפשר טיפול בכל אלפי המסמכים הקיימים במאגר הנתונים. בכדי לאפשר קטלוג של כלל המסמכים הקיימים ברשת, בחרנו לבנות מערכת המשתמשת באלגוריתם המבצע קטגוריזציה של טקסט בצורה אוטומטית.

קטלוג טקסט אוטומטי דורש להסיק מתוכן הדף – הכתוב בשפה טבעית- מהו נושאו המרכזי של המסמך, ומהי הקטגוריה המתאימה ביותר, מתוך קבוצת הקטגוריות הקיימת, לכלול את המסמך תחתיה. זאת, ללא כל מידע נלווה, פרט לתוכן המסמך עצמו. קטלוג שכזה יאפשר לאפיין את כלל המסמכים הקיימים במאגר המידע של מנוע החיפוש, תחת קטגוריה כל שהיא, ובכך לאפשר להציג תוצאותיו של כל חיפוש באמצעות קטגוריות, אשר יקלו על המשתמש לאתר את מבוקשו.

1.2. המטרה הממוקדת של הפרויקט

מטרתו העיקרית של פרויקט זה היא בניית מערכת, המתבססת על מנוע חיפוש קיים, אשר מאפשרת שליפת מידע ממוקד ורלוונטי יותר לשאילתות חיפוש. עיקרי המערכת יחולקו לשניים:

1. פונקצית קטלוג טקסט אשר תפעל offline על מאגר המסמכים של מנוע החיפוש ותאפיין בצורה אוטומטית כל מסמך ע"י קטגוריה מתאימה.
2. ממשק אשר יתבסס על קטלוג זה של המסמכים במאגר ומאפייניו יהיו כדלהלן:

1. כבכל ממשק של מנוע חיפוש, תינתן למשתמש האפשרות להגיש שאילתת חיפוש. כמו כן, תינתן אפשרות לקבל תוצאות רק מקטגוריות מסוימות מתוך רשימה נתונה.
2. עיבוד השאילתה יתבצע ע"י מנוע החיפוש בצורה הרגילה.
3. תוצאות החיפוש שיתקבלו ממנוע החיפוש, יחולקו לקבוצות, כאשר מסמכים אשר קוטלגו ע"י האלגוריתם תחת אותה קטגוריה, יהיו באותה קבוצה.
4. בהצגה הראשונית יתאפשר למשתמש לראות את המסמכים אשר חזרו משאילתת החיפוש (כשם שמציגים מנועי החיפוש). אם בחר לקבל תוצאות רק מקטגוריות מסוימות, אזי יוצגו רק המסמכים מקטגוריות אלו.
5. המערכת תאפשר למשתמש לבחור את הקטגוריה (אחת או יותר אם יבחר) הרלוונטית לצרכיו, לאחר שיבחר יוצגו לו המסמכים שקוטלגו תחת קטגוריה זו.

פעולה זו תערוך סדר בתוצאות המתקבלות ממנוע החיפוש, 'תסנן' מסמכים רבים אשר אינם רלוונטיים לצורכי המשתמש, ותכוון המשתמש אל המסמכים הדרושים לו.

1.3. תרומת הפרויקט לפתרון הבעיה הממוקדת

פרוייקט זה נועד לתת מענה לבעייתיות הקיימת כיום בשליפת מידע ממערכות אחזור מידע גדולות. תרומתו, בהיותו כלי עזר להתמודדות עם היקף מסמכים נרחב המתקבל כתוצאה מהצגת שאילתא למנוע חיפוש קיים, ולשיפור ושכלול הדרכים הקיימות לאיתור מידע רלוונטי תוך זמן קצר. הגדרת הפרוייקט ואפיון מרכיביו נגזרים מן הבעיה שהוגדרה לעיל.

2. רקע תיאורטי

2.1. ממשק ככלי לסינון מידע

ממשק משתמש בא לענות על מספר צרכים. הצורך החיוני ביותר עליו הוא עונה הוא הצורך בקיומו של תווך בין האדם - מפעיל המערכת, לבין תוכניות המחשב המפעילות את מערכת המידע. אולם אנו מוצאים כי הממשק עונה על צורך נוסף, והוא - סינון מידע.

מלאכת איתור מידע ממוקד מתוך מאגר מידע טקסטואלי נרחב, דורשת עריכת סינון. ככל שמאגר המידע גדול יותר כך כמות התשובות, המתקבלות בתשובה על שאלת החיפוש, גדלה. אולם המשתמש מעוניין כי היקף התוצאות שיתקבל בידיו, לא יהיה רחב מידי וקריאתו (הראשונית) תתאפשר תוך פרק זמן סביר. היקפם הנרחב של מאגרי המידע הקיימים לא מאפשר לצמצם את מספר התשובות המתקבל בתגובה לשאילתת חיפוש, למספר סביר. עם זאת, ניתן לארגן את התוצאות המתקבלות בצורה נוחה ויעילה יותר. ממשק המשתמש יכול לשמש כלי עזר אפקטיבי לעריכת הארגון הדרוש.

הממשק בעיקרו, נותן בידי המשתמש יכולת סינון ובירור מהירה ויעילה תוך שימוש בכלים ויזואליים נהירים וקלים לשימוש. זאת, תוך מניעת בזבוז זמן מיותר על קריאת מסמכים שאינם רלוונטיים עבור המשתמש. [3]

2.2. אחזור מידע

2.2.1. רקע כללי

מאז שנות הארבעים, בעיית אחסון ושליפת מידע *(Information Retrieval)* מיקדה תשומת לב רבה. אולם הבעיה נותרה בעינה ואף החריפה עם התרחבות היקף המידע הנמצא בידי האנושות. בידנו כמות עצומה של מידע, בעוד מהירות הגישה אליו ומידת דיוקה הפכו להיות בעייתיות יותר ויותר. בעיה זו גורמת לעיתים, לשכפול המאמץ והעבודה.

לכאורה, אחסון מידע ושליפתו הנו דבר פשוט. חיפוש מידע ספציפי מתוך מאגר מידע גדול אפשר שיתבצע ע"י קריאת כלל המסמכים במאגר, תוך הבדלת המסמכים הרלוונטיים מאלו שאינם. כך נקיים את פעולת הסינון באופן "מעולה". עם זאת, ברור מאליו כי פתרון שכזה אינו מעשי, היות ולאדם התר אחר מידע ספציפי אין את הזמן, הרצון והיכולת לקרוא את כלל המסמכים הקיימים במאגרי המידע העצומים שלרשותו .

פיתוחם של מחשבים בעלי יכולת עיבוד מהירה, הנחיל באנשים סברה כי המחשב יהיה מסוגל 'לקרוא' אוסף שלם של מסמכים ולחלץ מתוכם את המסמכים הרלוונטיים. במהרה התברר כי לא רק ששימוש בשפה טבעית בטקסט המסמך גורם לבעיית קלט ואחסון, אלא גם נותרת 'בעיית חשיבה' לא פתורה של אפיון תוכן המסמך ע"י המחשב. אפיון אוטומטי של מסמך, אותו המחשב מעוניין לנסות ולממש, וכן ניסיון לשכפול תהליך אנושי של 'קריאה' ע"י המחשב – אינו מלאכה פשוטה כלל ועיקר. מלאכת ה - 'קריאה' מורכבת הן מניסיון לחלץ מידע – סמנטי וסינטקטי – מהטקסט והן מן השימוש בו להחלטה האם

המסמך רלוונטי או לא, לבקשה מסוימת. הקושי המחשבי אינו רק בידיעת מלאכת חילוץ המידע, אלא גם ובעיקר בהחלטה האם יש בו בכדי לתת מענה לבקשה הנדונה ואם לאו.

בחינת הרלוונטיות שבמידע לשאלה שהוגדרה, מהווה את לב תהליך שליפת המידע. מטרתה האסטרטגית של שליפת מידע אוטומטית הנה: שליפת כל המסמכים הרלוונטיים, בבת אחת, תוך פרק זמן קצר, כאשר כמות המסמכים הבלתי רלוונטיים, אשר בכל זאת נשלפו, תהיה קטנה ככל שניתן.

כאשר המחשב יאתר מסמך רלוונטי לבקשה שהוצגה, הוא יצרף המסמך לקובץ התוצאות המוחזר בתגובה לבקשה.[4]

2.2.2. מערכות אחזור מידע

Information Retrieval systems – מערכות שליפת מידע אמורות לבחור מסמכים שיש בהם עניין כלשהו למחפש. אולם גוף המסמך מכיל מילים, ולא רעיון, ולא תמיד יש התאמה בין המילים של המסמך, לרעיון. זה נובע מכך שלפעמים ישנן מילים שבשימוש מעבר לנושא אחד (כדוגמת המילים "עכבר" או "בנק"), ישנם רעיונות שדורשים לכינויים מעל למילה אחת, וכן יש המון רעיונות שיכולים להיות מתוארים מעבר למילה או לביטוי אחד. כדוגמת זוג המילים "doctor" ו - "physician" , ששניהם מתארים דוקטור.

בני אנוש מסיקים ומקישים את נושא המאמר ממילות המסמך, בצורה יחסית טובה. בשביל לעשות זאת הם משתמשים בעולמם, ומביאים גם את הידע העצום שלהם בדקדוק של אותה שפה. מעט מאוד ממידע זה זמין למערכת המחשב, ולכן נוצר המצב שיש לנו שיטות חסרות ולא שלמות, של ארגון והסקת מידע בצורה אוטומטית.

לעיתים קרובות, תוכנות המנתחות משפטים ומציגות את תוכנם הסמנטי בשפה פורמלית , נכשלות. אפילו במשימות יותר פשוטות, כמו ההחלטה האם שימושה הפרקטי של המילה "bear" צריך להיחשב כשם - עצם או כפועל, מערכות כאלה נכשלות.[5]

מערכת שליפת מידע מדומה בעצם לקופסא שחורה. איור 1 מתאר את שלושת מרכיביה: הקלט, המעבד והפלט.

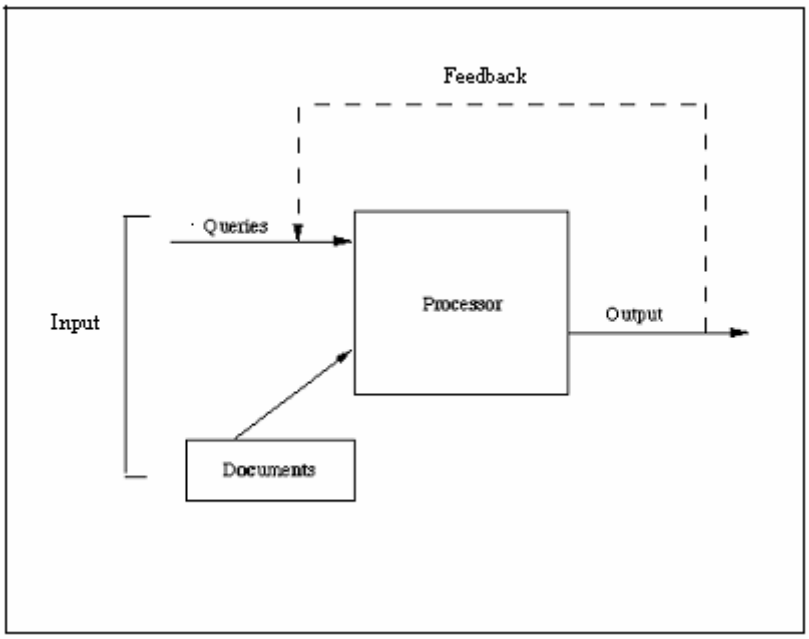
אם נתבונן בקלט נראה כי הוא מורכב ממאגר של מסמכים, אשר יש צורך במציאת דרך לייצג אותם בצורה כזו, שתתאים לשימוש המחשב, וכן משאילתת חיפוש. רוב מערכות שליפת מידע המבוססות מחשב מאחסנות רק את ייצוג המסמך. כלומר, טקסט המסמך נאבד ברגע העיבוד, ומה שנשמר מהמסמך הנו ייצוגו. ייצוג שכזה יכול להיות לדוגמא שמירת מילים משמעותיות מהמסמך.

לעומת מערכת העובדת בשפה הטבעית, יש אפשרות לעבוד עם שפה מלאכותית, אשר כל השאילתות והמסמכים מנוסחים בשפה זו. (כמובן בהנחה שהמשתמש מסכים ללמוד להביע את הצורך שלו במידע – באותה שפה).

אם נתבונן במעבד נראה כי חלק זה במערכת האחזור מתייחס לתהליך השליפה. התהליך יכול להכיל פונקצית שליפה, שמבצעת את אסטרטגיית החיפוש בתגובה לשאילתה, וכן בניית מידע בדרך מסוימת כמו סיווג ומיון הנתונים.

בדיאגרמה שבאיור 1, המסמכים מוקמו בקופסא נפרדת להעצים את העובדה שהם לא רק קלט, אלא יכולים גם להקל (בעזרת דרך הייצוג שלהם) בתהליך השליפה.

ולבסוף, נתבונן בפלט ונראה כי בדרך כלל זוהי קבוצת ציטוטים או מספר מסמכים. במערכת ביצועית הסיפור נגמר פה. אולם במערכת מחקרית, הוא מעביר זאת לביצוע הערכה. [4]



איור 1 - מערכת שליפת מידע אופיינית

2.2.3. מדדי יעילות

מטרתם של הרבה מחקרים ופיתוחים בשליפת מידע הנו שיפור ביצוע, יעילות ומהירות השליפה. יעילות נמדדת בד"כ במונחים של שימוש במשאבי המחשב כמו גיבוי האחסון וזמן C.P.U. . קשה למדוד יעילות במחשב בדרך בלתי תלויה. יעילות נמדדת בד"כ במדד *precision* ובמדד *recall*.

מדד *precision* הנו היחס של מס' המסמכים הרלוונטיים שחזרו בפעולת השליפה מול מספרם הכולל של כל המסמכים שחזרו בפעולה.

מדד *recall* הנו היחס של מס' המסמכים הרלוונטיים שחזרו בפעולת השליפה מול מספרם הכולל של המסמכים הרלוונטיים (מספר זה כולל את המסמכים שחזרו בפעולה וכן את אלו שלא חזרו, למרות שהנם רלוונטיים). [4]

2.3. מנועי חיפוש

2.3.1. סקירת מבנה הרשת

אתרי אינטרנט נמצאים במחשבי שרת ברחבי העולם. מחשבי שרת להבדיל ממחשבים אישיים מאפשרים ביקור וצפייה בתוכן שנמצא בהם, תכונה שאינה קיימת במחשבים אישיים המסוגלים לקבל מידע בלבד. מחשבי השרת מחוברים זה לזה בתקשורת מחשבים כמו רשת של קורי עכביש.

אתר באינטרנט מורכב מאוסף דפים שנקראים דפי web ונכתבים בשפת html. דפים אלה מכילים טקסט, קבצי תמונות, וידאו ו/או קול . אחד המאפיינים של דפי web הוא יכולתם לקשר מדף אחד לדפים אחרים באותו אתר או לאתרים אחרים - לינקים או קישורים.

2.3.2. הדרישות

יצירת מנוע חיפוש מעמידה מספר אתגרים לא פשוטים. דרושה טכנולוגיה שתאסוף את דפי הרשת ותדאג לעדכונם. דרוש שימוש יעיל במקום האחסון לשמירת אותם מסמכים. מערכת המפתח (indexing system) צריכה לעבד מאות ביליונים של בתים בצורה יעילה, וכן דרוש טיפול מהיר מאוד בשאילתות.

ככל שהרשת גדלה כך בעיה זו נהפכת להיות יותר ויותר קשה. אולם אנו מקבלים פיצוי חלקי מההתקדמות בנושא החומרה ועלותה. [6]

2.3.3. תוכנת רובוט

מנוע חיפוש הוא למעשה מחשב שרת המפעיל תוכנת רובוט (Robot). הרובוט סורק מחשבי שרת ברשת באופן שיטתי תוך מעבר מאתר לאתר על-פי הקישורים המופיעים באתר. כאשר הרובוט מגלה אתר חדש, הוא בודק אם הוא מורשה לתת לו מפתח. ואם כן, הוא מעתיק את כל תוכן האתר, או חלקו, אל השרת של המנוע, והמנוע ממפתח את המידע שנאסף. כל המילים בתוכן האתר נשמרות במסד הנתונים של המנוע. מלבד תוכן האתר, נשמרים במסד הנתונים גם נתונים נוספים כגון תאריך "הורדת" האתר, תאריך עדכון אחרון של האתר, תמצית אוטומטית קצרה, כותרת האתר ועוד.

בגלל שאתרי אינטרנט הם דינמיים, (מתעדכנים מזמן לזמן), רובוטים גם מעדכנים את האתרים שכבר קיבלו מפתח בעבר.

למרות שצורת הפעולה של כל המנועים דומה, לכל אחד מהם ייחוד משלו. הייחודיות היא באתרים שהרובוט סורק. יש רובוטים הסורקים כל אתר, ויש הסורקים רק אתרים בעלי מס' כניסות רב וקצב עדכון גבוה. ייחודיות המנוע היא גם גודל המאגר שלו. כן יש הבדלים במידע שהמנוע שומר בשרת שלו: יש מנועים השומרים את כל תוכן האתר, ויש השומרים רק את תחילתו (מתוך הנחה שתחילת האתר מעידה על נושאו) או את האתר עד "עומק" מסוים. דברים נוספים המבדילים בין המנועים הם מנגנון החיפוש וברירות החיפוש השונות, שיטת דירוג תוצאות החיפוש וחישוב רלוונטיות האתרים.

2.4. Clustering

דרך אפשרית לפתרון הבעיה, של רשימת תוצאות ארוכה החוזרת ממנוע החיפוש, נעזרת ב clustering - קיבוץ תוצאות החיפוש לקבוצות. ע"פ ההיפותזה עליה מסתמכת גישה זו, מסמכים רלוונטיים – המקיימים את תנאי החיפוש, בד"כ דומים יותר אחד לשני מאשר למסמכים לא רלוונטיים. על כן ניתן לקבץ מסמכים דומים לקבוצה אחת ניתנת לאפיון.

ישנן שתי אפשרויות – לביצוע clustering. האחת, לבצע pre-clustering על מאגר המסמכים. כאשר אנו עוסקים במנוע חיפוש אפשרות זו בעייתית, משום שכמות המסמכים היא רבה מאוד ודבר זה יארך זמן-רב. ולכן כאשר מדובר במנוע חיפוש, לרוב תיבחר האופציה של on-line clustering עבור כל שאילתת חיפוש. כאן ההתמודדות היא עם קבוצה קטנה בהרבה של מסמכים – קבוצת המסמכים הרלוונטיים, וניתן לחלק אותם לקבוצות ע"פ הדמיון בניהם.

2.5. קטלוג טקסט

בסעיף הקודם הצגנו אפשרות אחת להתמודד עם בעיית הצגת התוצאות. אפשרות אחרת היא קטלוג כל מסמך תחת קטגוריה, מתוך קבוצת קטגוריות מוגדרת. הקטלוג יתבצע off-line לכל מסמך. בזמן עיבוד שאילתת החיפוש, ניתן יהיה לקבץ מסמכים בעלי אותה הקטגוריה בצורה מהירה יותר, והקיבוץ יהיה ברמה גבוהה יותר. הסעיפים הבאים מציגים כיצד מתבצע קטלוג שכזה למסמכי טקסט.

2.5.1. הגדרה

ניתן להתייחס לבעיית קטלוג טקסט כאל השמה של ערכים בוליאניים לכל זוג

$<c_i, d_j> \in D \times C$, כאשר D הוא מרחב המסמכים ו- C היא קבוצה של קטגוריות

$C = \{c_1,...,c_{|c|}\}$, אשר מוגדרות מראש. זוג זה יקבל ערך T כאשר מסמך d_j יקוטלג תחת קטגוריה c_i, וערך F יהווה אינדיקציה לכך שהוחלט לא לסווג מסמך d_j תחת c_i.

אם כן המשימה בקטלוג טקסט היא בעצם להגדיר את פונקצית המטרה

$\Phi : D \times C \longrightarrow \{T,F\}$ שנקראת הפונקציה "המסווגת", אשר בעצם קובעת עבור כל מסמך d_j האם ניתן או לא ניתן לסווג אותו תחת הקטגוריה c_i.[9]

2.5.2. מסווג – Classifier

מסווג (classifier) הנה פונקציה המקבלת כקלט וקטורים של תכונות $\vec{X} = (x_1, x_2, ...x_n)$ ומשייכת אותם למחלקות (classes). כך שעבור כל וקטור היא מקבלת החלטה לאיזו מחלקה הוא שייך –

(class) *confidence* $=f(\vec{X})$. במקרה שלנו – text classifier, התכונות הן מילים במסמך, כך שכל וקטור מייצג מסמך, והמחלקות הן קבוצה נתונה של קטגוריות. המסווג, מסווג כל מסמך תחת הקטגוריה המתאימה. [7]

2.5.3. קטלוג תחת תווית אחת מול תוויות מרובות

במשימת הסיווג אנו יכולים לכפות אילוצים שונים, בסעיף זה נדון באילוץ של השמת k (בדיוק k , או לפחות k) איברים מהקבוצה C לכל מסמך מ D, ז"א נוכל לקבוע כי כל מסמך יקוטלג תחת k קטגוריות (או לפחות k קטגוריות) מתוך קבוצת הקטגוריות הנתונה. המקרה בו בדיוק קטגוריה אחת מושמת לכל מסמך מ – D , נקרא קטלוג תחת תווית אחת (נקרא גם המקרה הבינרי), והמקרה בו כל מספר קטגוריות בין $|C|$- 0 יכולות להיות מושמות לאותו מסמך נקרא מקרה תוויות מרובות.

אנו נעסוק במקרה הבינרי מהסיבות הבאות:

1 אם נפתור את המקרה הבינרי נוכל בקלות לפתור את המקרה הכללי.

2 הוא יותר מתאים לפתרון הבעיה שלנו.

3 רוב הספרות עוסקת במקרה זה.

אם כן אנו נתייחס לבעיית הקטלוג ל $C = \{c_1,...,c_{|c|}\}$ כאל $|C|$ בעיות בלתי תלויות של קטלוג מסמך d_j תחת קטגוריה, c_i לכל $1 \geq i \leq |C|$ ונתייחס אל הפונקציה Φ כאל פונקציה $\Phi_i : D \longrightarrow \{T,F\}$. [9]

2.5.4. גישות שונות בקטלוג

ישנן שתי אפשרויות לביצוע קטלוג של מסמכי טקסט:

1.לבדוק עבור כל מסמך לאיזה קטגוריה\ות הוא שייך
(DPC - document-pivoted categorization)

2.לבדוק עבור כל קטגוריה איזה מסמכים מקוטלגים תחתיה
(CPC - category-pivoted categorization)

ב DPC משתמשים כאשר בכל פעם מגיע לידינו מסמך חדש, ואנו רוצים לדעת תחת איזה קטגוריה לקטלג אותו.

ב CPC משתמשים כאשר רוצים לאפשר הוספה או הורדה של קטגוריה בצורה גמישה.

אנו נשתמש בגישת ה DPC, משום שבמקרה שלנו אנו נבדוק בכל פעם מסמך חדש ונרצה להחליט תחת איזה קטגוריה לקטלג אותו. [9]

2.6. קטלוג טקסט אוטומטי

2.6.1. מוטיבציה

קטלוג אוטומטי של מסמכי טקסט משמעותו – בניית תוכנה שמסוגלת לסווג מסמכי טקסט תחת קטגוריות מוגדרות מראש. לכלי שכזה יש כמובן יתרונות ושימושים רבים בתחומים שונים.

כפי שציינו בהגדרת הבעיה הפרטית, במקרה שלנו, כלי כזה יחסוך כוח אדם, זמן, כסף רב ויאפשר התמודדות עם כמות המסמכים העצומה הקיימת ברשת, וסיווגם.

2.6.2. ייצוג מסמך

כאשר ניגשים למשימת קטלוג טקסט, הצעד הראשון הוא להמיר מסמכים, שהם בד"כ מחרוזות של תווים, לתצוגה בצורה שמתאימה למשימת הקלסיפיקציה (לייצגם כוקטורים). בסעיפים הבאים נציג מספר אפשרויות לייצג מסמכי טקסט.

2.6.2.1. ייצוג בינארי

ככלל, מייצגים מסמך כווקטור של מספרים, כאשר מספרים אלו הנם המשקלות שניתנו למילים/לביטויים הנמצאים במסמך, ולכל מילה מתאימים מספר ייחודי – *id*.

ייצוג מסמך בייצוג בינארי הנו פשוט ונפוץ. נייצג מסמך ע"י מערך של מספרים, כאשר כל כניסה מתאימה ל – *id*, המייצג את המילה. בייצוג בינארי הופעת מילה במסמך תסומן במקום המתאים לה בווקטור באחד, ואי-הופעתה של המילה תסומן באפס – כלומר לא תיוצג באופן מפורש. מחישובי יעילות עולה כי עדיף שהוקטור יהיה ממוין.

לדוגמא: ניתן למילים הבאות *id* - (בואינג , 23) (טיסה , 5) (תעופה , 312) (משטרה , 9). אם בקובץ *d* יש רק את שלושת המילים הראשונות, אזי ייצוג בינארי של הקובץ *d* (ממוין) יראה כך:

5	23	312

2.6.2.2. ייצוג tfidf

ייצוג זה לעומת ייצוג בינארי מכיל עוד מידע – נותן 'משקל' לכל מילה, אך יכול להאט את תהליך העיבוד.

מדד *tfidf* מודד את המילה בצורה - "תדירות המילה במסמך ספציפי מול תדירות המילה בכלל המסמכים". כאשר אין הסתמכות רק על תוכן המסמך, אלא גם ניסיון לתת משקל תוך התחשבות באוסף המסמכים הקיים.

יהא N מספרם הכולל של המסמכים שכבר עובדו וכבר קבענו עבורם לאיזה קטגוריה הם משתייכים (ה-*Training set*). נגדיר את תדירות המילה w במסמך d – $tf(w)$: מספר הפעמים שהמילה w מופיעה במסמך d.

נגדיר – $df(w)$: מספר המסמכים שהמילה w מופיעה בהם.

אזי משקל *tfidf* של מילה w במסמך d: $$tfidf(w) = tf(w) \times \log\left(\frac{N}{df(w)}\right)$$

ניתן לראות כי מדד זה נותן משקל נמוך עבור מילים נפוצות, ומשקל גבוה עבור מילים נדירות.
כך, אם מילה מופיע במסמך כלשהו, אך מופיע גם בהרבה מסמכים אחרים, היא תקבל משקל נמוך שמשמעותו בעצם היא שמילה זו מאפיינת את המסמך הזה בצורה 'חלשה'. לעומת זאת אם היא מילה שאינה נפוצה ומופיעה במעט מסמכים, הסיכוי שהיא תאפיין את המסמך הספציפי שבו היא מופיעה, בצורה 'חזקה' גבוה יותר ולכן תקבל משקל גבוה.
כעת יש בידינו ווקטור, של משקולות *tfidf* המייצגים מילים במסמך, ווקטור אשר מייצג מסמך לא רק בצורה 'יבשה' – המילה מופיעה או לא מופיעה במסמך, אלא ווקטור שמאפיין את המסמך ע"י המושגים המשפיעים יותר והמשפיעים פחות על אופי המסמך. [9]

2.7. תהליך למידה אינדוקטיבי

2.7.1. מודלי קטלוג בסיסים

Machine Learning - מערכות לומדות, פותרות בעיות ע"י בחינת דוגמאות (פתורות).
כאשר רוצים לממש אפליקציות של מערכות אלו לקטלוג טקסט, יש להעביר את מסמכי הדוגמאות לאותה צורת ייצוג כמו של שאר המסמכים (שאינם משמשים דוגמאות).
כאשר עוסקים בקטלוג טקסט, יש להתייחס למספר תהליכים. תהליך ראשון יהיה עיבוד מוקדם של מסמכי הדוגמאות (אלו שכבר קיבלו קטגוריה – *Training set*) והעברתם לייצוג בצורה הרצויה (לדוגמא, לייצוג *tfidf*). התהליך הבא יהיה ייצוג המסמך וסיווגו; כלומר, סווג מסמך יחיד מאוסף המסמכים בעזרת ה-*Training set* וקישור המסמך לתווית. (כאשר תווית מזוהה באופן ייחודי עם הקטגוריה שלה). [5]

2.7.2. Training set and Test set

מערכות לומדות מסתמכות על כך שקיים אוסף התחלתי $\Omega = \{d_1,...,d_{|\Omega|}\} \subset D$ של מסמכים (קבוצה חלקית למרחב המסמכים D), אשר כבר קוטלגו תחת אחת הקטגוריות $\{c_1,...,c_{|C|}\} = C$. ז"א הערכים של הפונקציה $\Phi : D \times C \to \{T, F\}$ ידועים עבור כל זוג $\langle d_j, c_i \rangle \in \Omega \times C$, כאשר $\Omega \subset D$.
מסמך d_j הנו 'דוגמא חיובית' של קטגוריה c_i אם $\Phi(d_j, c_i) = T$, 'ודוגמא שלילית' של c_i אם $\Phi(d_j, c_i) = F$.

בדיקת אפקטיביות של מסווג כל שהוא $\breve{\Phi}$, נעשית ע"י חלוקת קבוצה זו של מסמכים, שכאמור אנו כבר יודעים את סיווגם, לשתי קבוצות אשר לא בהכרח שוות בגודלן:

- *Training set* $TV = \{d_1,...,d_{|TV|}\}$. קבוצה של מסמכים שהמסווג $\breve{\Phi}$ יודע מהם הערכים שנתן להם המסווג Φ, ובהם הוא ישתמש בתהליך הבנייה האינדוקטיבי.
- *Test set* $Te = \{d_{|TV|+1},...,d_{|\Omega|}\}$, קבוצת מסמכים זאת באה כדי לבדוק את האפקטיביות של המסווג $\breve{\Phi}$. כל מסמך בקבוצה זו יוזן למסווג $\breve{\Phi}$, והחלטות המסווג $\breve{\Phi}(d_j,c_i)$ יושוו עם החלטות המסווג 'המומחה' $\Phi(d_j,c_i)$. המדד לאפקטיביות של מסווג $\breve{\Phi}$, יבוסס על מספר ההתאמות בין הערכים $\breve{\Phi}(d_j,c_i)$ לערכים שנתן המסווג 'המומחה' $\Phi(d_j,c_i)$.

על מנת שבדיקה זאת תהיה אפקטיבית ובעלת הערכה מציאותית, וכדי להימנע ממצב בו נבחן את המסווג על אותם מסמכים שכבר ידוע מה קטלוגם (והיו שותפים בבנייתו), יש לדאוג כי המסמכים ב – *Te* לא משתתפים בבנייה האינדוקטיבית של המסווגים.

כעת לאחר שבוצעה הערכה, ניתן להריץ את המסווג על כל הקובץ ההתחלתי, על מנת 'לאמן' אותו ולשפר את ביצועיו. גישה זאת נקראת גישת "*train-and-test*" .[9]

2.8. מסווגים

2.8.1. מסווג K-nn

מסווג זה מתבסס על הימצאותה של קבוצת מסמכים התחלתית (*Training set*), אשר כל מסמך בקבוצה קוטלג ע"י מומחים, תחת קטגוריה מתוך קבוצת קטגוריות נתונה. בגישת ה K-NN כאשר המסווג נדרש להחליט, עבור מסמך חדש d_j, תחת איזו קטגוריה c_i מתאים יותר לקטלג אותו, הוא בוחן את הקטלוג של k המסמכים (מתוך ה *Training set*) הדומים ביותר ל d_j .

מסמך מיוצג ע"י וקטור של משקולות *tfidf* שנקבע כפי שהצגנו בסעיף 2.6.2.2 .

הדרך לבדוק את הדמיון בין המסמך החדש לבין מסמך מתוך ה *Training set* , היא לחשב את המרחק האוקלידי או ערך הקוסינוס (cosine) בין הוקטורים שמייצגים את שני המסמכים.

הדמיון לפי הקוסינוס נמדד ע"פ הנוסחה הבאה[8]:

$$Sim(X,\ D_j) = \frac{\sum_{t_i \in (X, D_j)} x_i \times d_{ij}}{\|X\|_2 \times \|D_j\|_2}$$

כאשר X הוא המסמך החדש, (מיוצג כוקטור), D_j הוא המסמך ה j-י ב- *Training set*, t_i היא מילה אשר נמצאת גם ב- X וגם ב- D_j, x_i הוא משקל המילה t_i ב- X, ו- d_{ij} הוא משקלה של t_i במסמך D_j. $\|X\|_2 = \sqrt{x_1^2 + x_2^2} + \ldots$ הנורמה של X, ו $\|D_j\|_2$ הנורמה של D_j.

כעת, לאחר שבידנו מדד לדמיון בין המסמך אותו אנו מבקשים לקטלג d_j, לבין כל מסמך ב *Training set*, ניתן למצוא את k המסמכים הדומים אליו ביותר.

השלב הבא, הוא להחליט תחת איזו קטגוריה c_i לקטלג את d_j. החלטה זו נעשית ע"י חישוב הערך CSV_i עבור כל קטגוריה c_i ($1 \geq i \leq |C|$). d_j יקוטלג תחת הקטגוריה אשר עבורה ערך זה הוא המקסימלי.

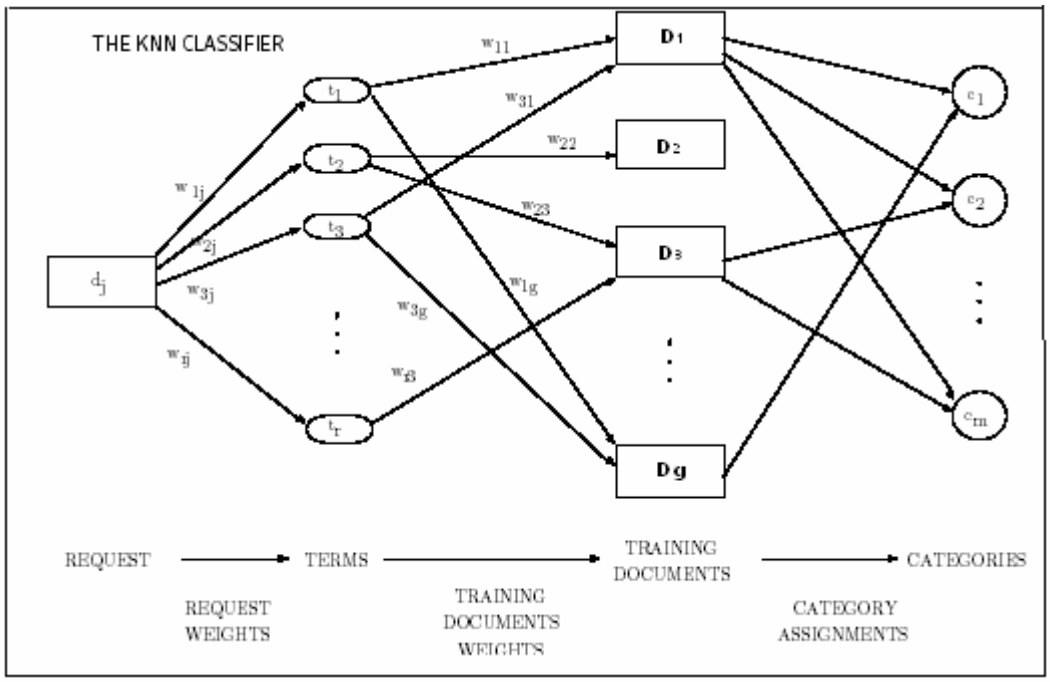
CSV_i מחושב בצורה הבאה:

$$csv_i = \sum_{D_z \in Tr_k(d_j)} Sim(d_j, D_z) \cdot \Phi(D_z, c_i)$$

כאשר $Tr_k(d_j)$ היא קבוצת k המסמכים הדומים ל d_j מתוך ה *Training set*. הפונקציה Φ, כפי שהסברנו קודם, היא ערך שניתן ע"י מומחה –ערך 0 או 1, שמשמעו D_z קוטלג ע"י המומחה תחת c_i, או לא.

נשים לב, כי ע"פ מדד זה, ככל שהמסמך D_z דומה יותר למסמך החדש d_j, כך הוא משפיע יותר. הקטגוריה בעלת ערך CSV_i הגבוה ביותר, תיבחר ע"י האלגוריתם כקטגוריה המתאימה ביותר לתאר את d_j, והוא יקוטלג תחתה.

איור 2 מציג בצורה גרפית את דרך העבודה של מסווג זה. הצד השמאלי של האיור, מראה את הקשר והדמיון בין המסמך d_j לכל אחד מהמסמכים ב *Training set*, וזאת ע"פ המשקולות(w_{ij}) שכל אחד מהם נתן למושגים (t_i) משותפים. בצד ימין רואים את הקטגוריות (c_i) של k המסמכים הדומים ביותר. משקל כל קטגוריה נקבע ע"פ ה CSV_i שלה, הקטגוריה שתיבחר בסוף התהליך, היא זו בעלת המשקל הגבוה ביותר. [9]



איור 2 - תצוגה גרפית של k-nn

2.8.2. מסווג BoosTexter[10]

במסווג זה נשתמש במערכת שלנו, נסביר בקצרה את פעולתו.

BoosTexter , הנו מסווג טקסט המבצע תהליך של למידה, כך שלאחר תהליך למידה זה יהיה מסוגל לבצע סיווג של מסמך חדש. גם אלגוריתם זה, מתבסס על קבוצת מסמכים התחלתית, אשר אנו יודעים כבר את קטלוגם לפי קטגוריות שנקבעו מראש וכן על קבוצת קטגוריות נתונה. קבוצת מסמכים זו (Training set) משמשת את תהליך הלמידה בבניית חוק קלסיפיקציה מדויק. הרעיון המרכזי של תהליך ה – Boosting הוא צירוף של חוקי קלסיפיקציה פשוטים ולא מדויקים, המייצגים היפותזות "חלשות" , לחוק אחד בודד – היפותזה אחת, מדויקת. היפותזה זו תאפשר למסווג לקבוע עבור מסמך חדש ולא מסווג, את הקטגוריה המתאימה ביותר מתוך קבוצת הקטגוריות הנתונה. האלגוריתם יבצע דירוג של הקטגוריות לפי התאמתן למסמך, ומכיוון שאנו מעונינים להתאים רק קטגוריה אחת לכל מסמך נתון, הקטגוריה שתיבחר היא זו שקיבלה את הדרוג הגבוה ביותר.

אלגוריתם ה – Boosting מסתמך על קיומה של פונקציה, אשר את דרך פעולתה נציג בהמשך, הנקראת– *Weak Learner* (*WL*) , אשר בכל קריאה לה, מייצרת חוק קלסיפיקציה פשוט. האלגוריתם, קורא לפונקציה זו בלולאה, כאשר לאחר כל קריאה, מצורף החוק שנוצר, לקבוצה של חוקים פשוטים, קבוצה של היפותזות חלשות. לאחר סיום הלולאה, מצרף האלגוריתם את כל ההיפותזות החלשות, לחוק אחד סופי – להיפותזה אחת חזקה ומדויקת. היפותזה זו נשמרת ע"י ה- BoosTexter בקובץ בינרי, קובץ זה ישמש אותו מעתה והלאה במשימת הקלסיפיקציה.

כעת, כאשר נרצה שה- BoosTexter יקטלג עבורנו באופן אוטומטי מסמך חדש, ניתן לו כקלט את המסמך ואת קובץ ההיפותזה. ע"פ ההיפותזה, אשר מייצגת בעצם את המידע שצבר ה- BoosTexter בתהליך הלמידה, יעבד האלגוריתם את המסמך החדש וייתן לנו דירוג של שייכות המסמך לכל אחת מהקטגוריות. אנו נבחר בקטגוריה בעלת הדרגה הגבוהה ביותר, ונקטלג את המסמך תחת קטגוריה זו.

נחזור כעת לפעולת ה- *Weak-Learner*. הפעולה הבסיסית של ה – *WL* דומה לעץ החלטה בעל רמה אחת. הבדיקה שמתבצעת בשורש העץ היא בסה"כ בדיקה, האם מושג כלשהו מופיע או לא מופיע במסמך הנתון. כל המילים או זוגות של מילים שכנות נחשבות למושגים. בהתבסס על התוצאה של בדיקה זו מפיק ה – *WL* חיזוי הקשור למושג זה. לדוגמא – מושג אפשרי יכול להיות למשל Clinton. והחיזוי האפשרי של ה – *WL* יכול להיות למשל: "אם המונח מופיע במסמך כל שהוא, אזי מסמך זה שייך לקטגוריה *חדשות* בסיכוי גבוה, שייך לקטגוריה *כלכלה* בסיכוי נמוך, ואינו שייך לקטגוריה *ספורט* בסיכוי גבוה.

לאחר בדיקת כל המושגים האפשריים, ובחירת חיזוי כזה לכל אחד מהם, נאספים כל החיזויים לידי חיזוי אחד $h(x,l)$ (ערך ההחזרה של ה- *WL*) המייצג את ההיפותזה לגבי שייכותו או אי שייכותו של המסמך x לקטגוריה l , (סדר הגודל $|h(x,l)|$ מייצג את רמת הביטחון של ה- *WL* לגבי שייכותו של המסמך לקטגוריה זו).

כפי שהסברנו קודם, ה- *WL* נקרא בלולאה. בסוף כל לולאה בודק אלגוריתם ה- Boosting את הפלט של ה- *WL* ומשווה אותו עם הקטגוריה האמיתית שניתנה למסמך. בריצה הבאה של הלולאה – בקריאה הבאה ל- *WL* יבקש ממנו האלגוריתם להתרכז ולתת חיזוי מדויק יותר לאותם מסמכים בהם החיזוי שלו היה רחוק מהמציאות, ז"א באותם המסמכים אשר ה- *WL* התקשה לקטלג אותם נכונה ונתן לקטגוריה הנכונה רמת בטחון נמוכה.

בצורה כזו מבצע האלגוריתם את תהליך הלמידה שלו שבסופו ידע כיצד משפיע כל מושג על קביעת הקטגוריה של מסמך, ומהם המושגים המשפיעים יותר והמשפיעים פחות על קביעת הקטגוריה המתאימה.

2.8.2.1. **שימוש בתכנה- BoosTexter**

כפי שאמרנו, אנו נשתמש במסווג BoosTexter המבצע את אלגוריתם ה- Boosting שתיארנו. תכנה זו, שנכתבה ע"י Y. Singer ו R. Schapire מצויה ברשת ומותרת לשימוש למטרות לימודיות וניסוייות.

בעקרון, תכנה זו מיועדת להפעלה על מערכות UNIX. אך מכיוון שסביבת העבודה שלנו היא WINDOWS עקב השימוש ב- Index Service ובשרת IIS, אנו נריץ את ה- BoosTexter על WINDOWS. ההרצה בסביבת WINDOWS תיעשה ע"י שימוש בתוכנה UWIN, המדמה Shell של UNIX על WINDOWS.

הרצת ה- BoosTexter נעשית בשני מצבים שונים:

- הראשון הוא המצב של תהליך הלמידה של האלגוריתם.
- השני הוא הרצת ה- BoosTexter כמסווג.

במצב הראשון, הקלט הוא שני קבצים. קובץ המתאר את כל מסמכי ה- Train-set כאשר בסוף כל מסמך מצוינת הקטגוריה שלו (הסיומת של קובץ זה הוא DATA). וכן קובץ המתאר את קבוצת הקטגוריות האפשריות בסיווג זה. הפלט של מצב זה הוא קובץ שהסיומת שלו היא SHYP (Strong-HYPotheses). קובץ בינרי זה, מתאר את ההיפותזה שיצר האלגוריתם בתהליך הלמידה. כאשר התחיליות של קבצים אלה צריכות להיות זהות.

הקלט של מצב ההרצה השני הוא קובץ ההיפותזה שנוצר קודם וקובץ המסמך שאנו מעונינים לסווג. האלגוריתם מבצע קלסיפיקציה של מסמך זה, ומחזיר כפלט קובץ הנותן דרוג לכל קטגוריה אפשרית, ביחס למסמך. הקטגוריה שנבחר לקטלג תחתה את המסמך היא זו בעלת הדרוג הגבוה ביותר.

.2.8.2.2 רמת הדיוק של האלגוריתם

ישנם שני פרמטרים מרכזיים המשפיעים על רמת הדיוק של אלגוריתם זה:

1. מספר המסמכים ב – Training-set. ככל שהאלגוריתם 'יתאמן' על מספר מסמכים רב יותר, כך גדל הסיכוי שתהליך הלמידה יהיה אפקטיבי יותר, ותוצאות הקלסיפיקציה יהיו מדויקות יותר.
2. אורך הלולאה. ככל שנריץ את הלולאה יותר פעמים, הסיכוי שהחיזוי יהיה מדויק, גדול יותר.

הערה: השפעתו של פרמטר זה, תלויה גם בפרמטר הקודם. אם למשל מספר המסמכים ב- Training-set יהיה קטן, לאחר מספר פעמים שנריץ את הלולאה, האפקטיביות של ההרצה תרד, ולא יהיה שינוי משמעותי בחיזוי ובדיוק.

את הערכת השגיאה נציג ע"פ נתונים שפרסמו כותבי התוכנה, לאחר בדיקות שבוצעו על ידם.

נגדיר מסווג H , H: $X \to Y$. H מתאימה למסמך $x \in X$, קטגוריה (תווית) $y \in Y$ בצורה הבאה:

$$\mathrm{H}(x) = max \text{ rank } \; l \in y \;\; f(x, y)$$

כאשר הפונקציה f, היא בעצם ה- BoosTexter אשר ביחס לכל מסמך נותן דירוג לכל אחת מהקטגוריות. השימוש שאנו עושים ב- BoosTexter, זהה לשימוש שהמסווג H עושה בו. H בוחר לסווג כל מסמך, תחת הקטגוריה שדורגה, ע"י f, הכי גבוהה ביחס למסמך זה.

נגדיר את קבוצת ה- Training-set - S

$$\mathrm{S} = \langle (x_1, Y_1), \ldots, (x_m, Y_m) \rangle$$

אזי הערכת השגיאה היא:

$$\text{one-err}_S(H) = \frac{1}{m}\sum_{i=1}^{m}[\![H(x_i) \neq Y_i]\!]$$

הערכה זו, מודדת את שגיאת האלגוריתם, כיחס של מספר המסמכים שלא קוטלגו נכונה ע"י H, ביחס למספר המסמכים ב- Training-set (m).

לא קוטלגו נכונה משמע, שהקטגוריה שנתנה להם ע"י H (הקטגוריה בעלת הדרגה הגבוהה ביותר), שונה מהקטגוריה הנכונה למסמך זה כפי שמופיע ב- Training-set.

ניתן לראות, כי בנוסחה זו משתקפים שני הפרמטרים המשפיעים על השגיאה, כפי שצוין קודם.

1. אורך הלולאה- ככל שנריץ את הלולאה יותר פעמים, הסיכוי שהקטגוריה הנכונה תקבל את הדרוג הגבוהה ותבחר ע"י H כקטגוריה הנכונה, גבוה יותר ולכן הסיכוי ש $H(x_i) \neq Y_i$ יקטן.
2. מספר המסמכים ב- Training-set - m , משפיע גם הוא על השגיאה. ככל ש- m גדול יותר כך השגיאה קטנה יותר.

3. תכנון המערכת

3.1. שלבי המערכת העיקריים

המערכת תפעל בארבעה שלבים עיקריים:

- פיתוח שאילתת חיפוש
- עיבוד השאילתה
- ארגון המסמכים
- הצגת התוצאות

חלק נוסף של המערכת, הוא ה Classifier, אשר עובד כל העת מאחורי הקלעים, על מאגר המסמכים של מנוע החיפוש, ותומך בפעולת המערכת.

3.1.1. פיתוח שאילתת חיפוש

ממשק HTML יאפשר למשתמש להזין שאילתת חיפוש, בצורה הבאה:

- הזנת פרטי השאילתה, בדומה לכל מנוע חיפוש.
- אפשרות לבחור, האם להציג מסמכים רק מקטגוריות נבחרות (אפשרות לבחור קטגוריות מתוך רשימה נתונה), או לקבל את עץ הניווט של הקטגוריות, שהתקבל עבור שאילתה זו.

3.1.2. עיבוד השאילתה

תהליך עיבוד השאילתה יעשה ע"י ה- Index Service .

Index Service הנה תוכנה המקבלת ספרייה מקומית (קטלוג), בונה לה טבלת אינדקס (באופן חד-פעמי), ועבור כל שאילתה ניגשת לאינדקס ומחזירה לאחר חיפוש את אותם מסמכים העונים לבקשה.

הרשומות של המסמכים שמקיימים את תנאי החיפוש, יועברו כפלט לאובייקט שאחראי על ארגון המסמכים.

3.1.3. ארגון המסמכים

ארגון תוצאות החיפוש ע"י המערכת יתבצע בשלבים הבאים:

- עבור כל מסמך שחזר משאילתת החיפוש, נשלוף מהטבלה את הקטגוריה שהותאמה לו.
- נתוני המסמכים יישמרו בקובץ xml (Extensible Markup Language).

הממשק ישתמש בקובץ זה, ע"מ ליצור הצגה של תוצאות החיפוש בצורה מסודרת ויעילה.

3.1.4. הצגת התוצאות

כעת כשבידי הממשק, תוצאות החיפוש מקובצות לפי קטגוריות, הן יוצגו למשתמש בצורה הבאה:

- אם המשתמש בחר לקבל תוצאות מקטגוריות מסוימות, יוצג עץ ניווט רק עבור קטגוריות אלו (אם חזרו מסמכים שקוטלגו תחת קטגוריות אלה).
- אם לא בחר באפשרות זו, יוצג עץ ניווט, אשר יאפשר למשתמש, בכל שלב, לראות מסמכים מקטגוריה כלשהי, או מאחת מתת הקטגוריות שלה.

3.2. מרכיבי המערכת

למערכת שנבנה שני חלקים מרכזיים, הסעיפים הבאים ידונו בפרטים של כל אחד מהם.

3.2.1. מנוע חיפוש

מאחורי הקלעים של המערכת שלנו ישב מנוע חיפוש, אשר יעבד את שאילתת החיפוש, וייתן כפלט לממשק שנבנה, את המסמכים שיקיימו את תנאי החיפוש.

מכיוון שאין בידי המכללה אפשרות לאפשר לנו התקנת מנוע חיפוש ולהתממשק אליו, אנו נאלץ לבנות את המערכת תחת מספר מגבלות. אנו נדמה את פעולת מנוע החיפוש בעזרת Index Service של Microsoft. כלי זה מאפשר לבנות אינדקס למסמכים מקומיים הנשמרים תחת ספריה מוגדרת מראש (הנקראת קטלוג), ומאפשר ביצוע חיפוש, בעזרת שאילתת חיפוש בדומה למנוע חיפוש רגיל, במאגר מסמכים זה.

3.2.1.1. מאגר מסמכים

כל מנוע חיפוש מחזיק מאגר של מסמכים שהוא אסף מהרשת. עבור כל מסמך במאגר הוא מחזיק מספר פרמטרים המאפיינים את המסמך. ה Index service , מחזיק עבור כל מסמך את המאפיינים הבאים:

- שם המסמך.
- גודל המסמך.
- תאריך שינוי אחרון.
- מיקום המסמך.

#	Title	Size	Modified	Path
1.	readme.txt	4981	3/20/3	c:\nvidia\winxp-2k\43.45\readme.txt
2.	ie.txt	2886	7/23/1	c:\windows update setup files\ie.txt

איור 3 – הנתונים אותם מחזיק ה-Index Service עבור כל מסמך

באיור 3 Readme.txt ie.txt הם דוגמאות למסמכים בקטלוג של ה Index service, כאשר עבור כל אחד מהם מוחזקים הפרמטרים שציינו.

אנו נרצה להוסיף עבור כל רשומה שבמאגר (רשומה מייצגת מסמך), מאפיין נוסף (כלומר עמודה נוספת), והוא קטגוריה אליה משתייך מסמך זה. דבר זה יעזור לנו בהמשך, בהצגת התוצאות למשתמש.

3.2.1.2. אינדקס

ככל מנוע חיפוש, גם Index Service בונה אינדקס עבור המסמכים שברשותו. אינדקס זה מאפשר לעבד את שאילתת החיפוש בצורה מהירה ויעילה.

3.2.1.3. Organizer

ה Organizer מקבל מה- Index Service את תוצאות החיפוש. עבור כל אחד מהמסמכים, הוא שולף את הקטגוריה תחתיה הוא קוטלג, ממסד הנתונים – ושומר את נתוני המסמכים בדף XML.

3.2.1.4. Classifier

ה-Classifier יסווג כל מסמך במאגר המסמכים של מנוע החיפוש (כל מסמך שעדיין לא סווג), תחת קטגוריה, מתוך קבוצת קטגוריות נתונה מראש.

המסווג, ירוץ באצווה (batch), מאחורי הקלעים, על מאגר המסמכים. דבר אשר יחסוך זמן, בזמן ביצוע עיבוד של שאילתה ע"י מנוע החיפוש.

3.2.2. ממשק

3.2.2.1. הכנסת קלט

הכנסת הקלט תתבצע בעזרת ממשק HTML , אשר כמו בממשקים של רוב מנועי החיפוש, יאפשר למשתמש להקליד שאילתת חיפוש. כמו כן תינתן למשתמש האפשרות לבחור בין שתי האופציות הבאות:

- לקבל את הפלט כעץ ניווט (שיוסבר בהמשך).
- לקבוע מראש מאילו קטגוריות, מתוך קבוצת קטגוריות, יוצגו המסמכים שחזרו מפעולת החיפוש.

באיור 4 ניתן לראות כיצד יראה ממשק הקלט.

איור 4 – ממשק הקלט

3.2.2.2. הצגת הפלט

שאילתת החיפוש תעובד ע"י המערכת ותוצאות החיפוש יוצגו בצורה הבאה:

אם המשתמש בחר באפשרות השנייה (מהסעיף הקודם), אזי יוצגו לו מסמכים רק מהקטגוריות הנבחרות.

בכל מקרה, התוצאות יוצגו בצורה של עץ ניווט (בדומה לWindows Explore). כלומר, לא יוצגו כל המסמכים שחזרו משאילתת החיפוש בצורה סדרתית, אלא בכל שלב תינתן לו האפשרות לראות את כל המסמכים מקטגוריה כלשהי בעץ הקטגוריות, או לרדת בעץ הקטגוריות לאחת מתוך תת הקטגוריות, ולהתמקד רק במסמכים הלקוחים מתוך תת קטגוריה זו.

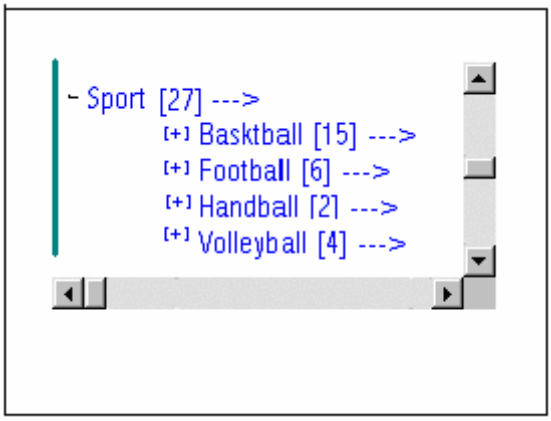
לדוגמא, נניח כי בעץ הקטגוריות שלנו שתי רמות בלבד. הקטגוריה הראשית – קטגוריית השורש היא ספורט, ותת הקטגוריות שלה הן: כדורסל, כדורגל, כדוריד, כדורעף וכדור מים. אזי, כפי שמתואר באיור 5 עץ הניווט של תוצאות החיפוש יראה, לדוגמא, כך:

סה"כ חזרו 27 מסמכים משאילתת החיפוש - Sport [27] . כעת יוכל המשתמש לבחור האם לראות את כל רשימת המסמכים – ע"י לחיצה על sport. או לראות את תת הקטגוריות של sport ע"י לחיצה על + שמשמאל לקטגוריה (כשם שמוצג באיור 5). לאחר שבחר באפשרות זו, ייפתח תת העץ של הקטגוריה הנבחרת, ויוצגו תת הקטגוריות של הקטגוריה, אך רק אותן תת קטגוריות מהן חזר לפחות מסמך אחד.

בדוגמא שלנו אנו רואים כי הופיע תת העץ של sport, אך נעדרת ממנו הקטגוריה כדור מים משום שלא חזרו מסמכים שקוטלגו תחת קטגוריה זו.

כך יוכל המשתמש בכל שלב להתמקד בנושא אותו הוא מחפש, ולצמצם את כמות המסמכים המוחזרים, וכל זאת יעשה ללא צורך בעיבוד השאילתה מחדש.

כמו ב Windows Explore, ניתן יהיה להתקדם בעץ בצורה מקבילית, לעלות רמות וכו'.



איור 5 – עץ ניווט

3.3. מערכת הפעלה ושפות תכנות

מערכת ההפעלה בה נשתמש הנה Windows 2000 Professional. זאת מכיוון שאנו עושים שימוש בתוכנת Index Service כמנוע חיפוש. כמו כן אנו עושים שימוש בשרת IIS (Internet Information Services) של Microsoft. רכיבים אלה נתמכים ע"י מערכת הפעלה זו.

השפות בהן נשתמש הן:

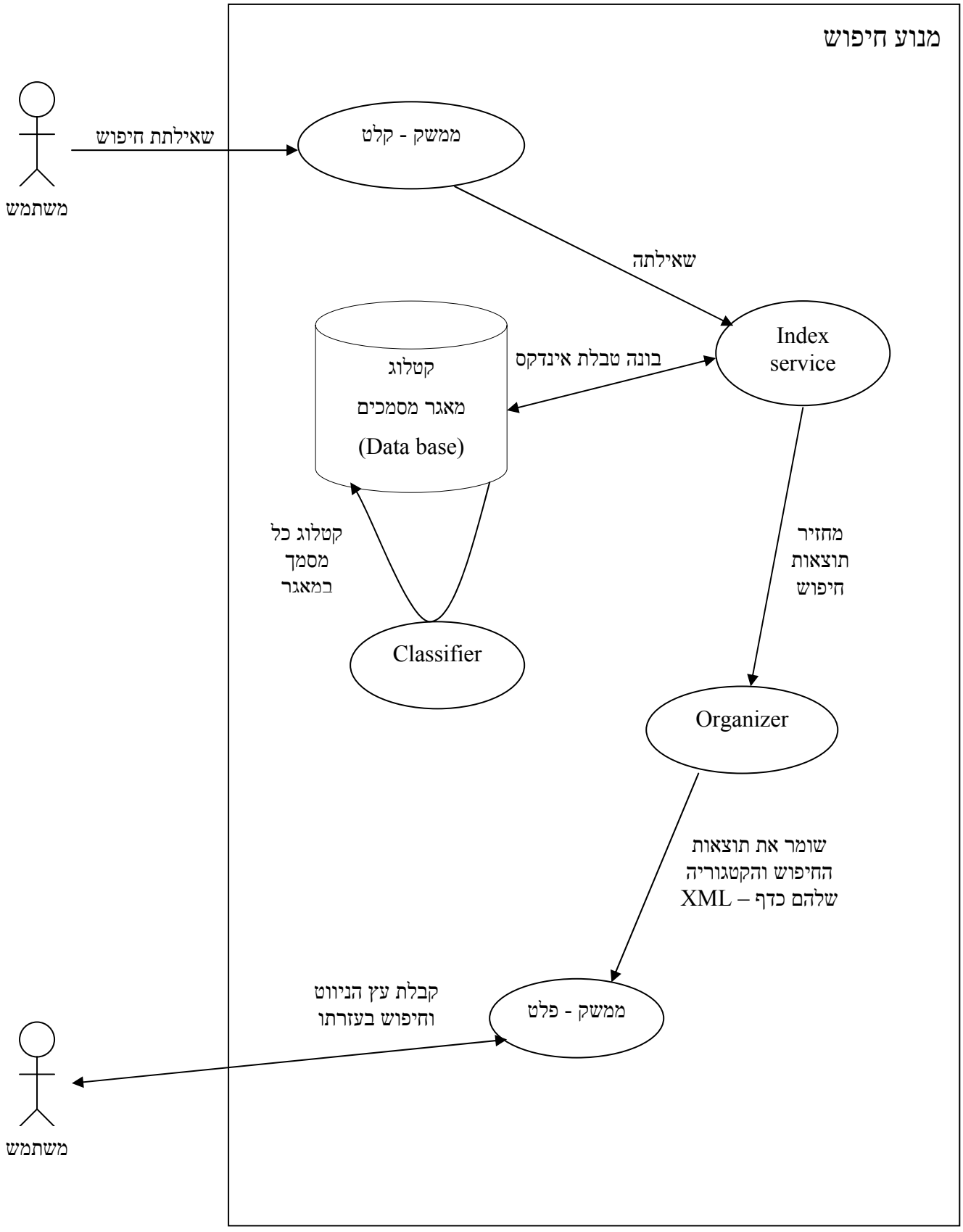
צד השרת ייכתב ב- ASP (Active Server Pages) .

ארגון מאגר המסמכים ייכתב ב- Java.

בבניית הממשק, צד השרת ייכתב ב-VBScript (Visual Basic Script) וצד הלקוח ב- JavaScript.

בניית דפי התוצאות תעשה בעזרת XML - לשם כך נצטרך IE-5 (Internet-Explorer 5).

3.4. דיאגרמת מבנה המערכת



4. המערכת הממוחשבת שנבנתה בפועל

למערכת שני חלקים עיקריים - ממשק משתמש אינטראקטיבי ותהליך קטלוג.

תהליך הקטלוג אחראי על קטלוג המסמכים במאגר, ובניית מבני הנתונים בהם ישתמש הממשק. בעוד הממשק מאפשר הכנסת קלט, ואחראי על הצגת פלט השאילתה.

4.1. מבני נתונים

המערכת עושה שימוש בשני מבני נתונים עיקריים:

- ספריית ה- DataBase : ClassifiedDocumentsDataBase – ספרייה אשר בה מאוחסנים פיזית המסמכים עליהם אנו מאפשרים ביצוע חיפוש.
- מסד הנתונים : DocumentsCategories - מסד נתונים טבלאי, אשר מכיל רשומה עבור כל מסמך מה – DataBase.

כמו כן המערכת משתמשת בספרייה נוספת :

- ספריית UnClassifiedDocumentsDataBase – אשר אליה מוריד ה"עכביש" של מנוע החיפוש קבצים מהרשת (כל מנוע חיפוש מכיל תוכנה כזו – לא מומשה על ידינו). קבצים אלה נשמרים בספרייה זו עד שנקבעת להם קטגוריה ע"י המקטלג. לאחר שנקבעת להם קטגוריה הם מועברים מספרייה זו לספריית ה- ClassifiedDocumentsDataBase ורק אז הם משתתפים כחלק פעיל במערכת (התהליך כולו יוסבר בפרוט בהמשך).

מבני נתונים אלו נבנים ומתעדכנים ע"י תהליך הקטלוג, ומשמשים את הממשק האינטראקטיבי.

4.1.1. ספריית ה- Database

כפי שצוין, ספרייה זו (ClassifiedDocumentsDataBase) מאחסנת פיזית את המסמכים שמרכיבים את מאגר הנתונים של המערכת. ספרייה זו מוגדרת להיות ה"קטלוג" של ה- Index Service. המשמעות של הגדרה זו, שה- Index Service מבצע חיפוש של ערכי שאילתה, רק בתוך הקבצים שנשמרים תחת ספרייה זו (או תתי - ספריות שלה). ה- Index Service בונה index לקבצים בספרייה, דבר המאפשר לו לעבד שאילתה בצורה מהירה יותר.

ספרייה זו ובמקביל לה גם הטבלה ClassifiedDocumentsTable (תתואר בהמשך בסעיף 4.1.2.1) מתעדכנות מעת לעת (כפי שיוסבר בהמשך בעיצוב התוכנה) בקבצים חדשים שהורדו מהרשת ועברו את התהליך של קביעת קטגוריה ע"י המקטלג.

4.1.2. מסד הנתונים - DocumentsCategories

מסד נתונים זה (מסד נתונים מסוג Access) מכיל שתי טבלאות : טבלת ClassifiedDocumentsTable, שהנה טבלה קבועה וטבלת UnClassifiedDocumentsTable שהנה טבלה זמנית.

4.1.2.1. טבלת ClassifiedDocumentsTable

בטבלת ה- ClassifiedDocumentsTable נשמרת רשומה עבור כל מסמך מה- Data-Base . השדות של כל רשומה הם:

- documentPath – מיקומו הפיזי של הקובץ בשרת. שדה זה הוא גם המפתח של הטבלה משום שהוא לבטח ייחודי לכל קובץ. שליפת נתונים מהטבלה תתבצע ע"פ שדה זה, דבר אשר יאפשר שליפה מהירה יותר.
- documentCategory - הקטגוריה שניתנה למסמך זה ע"י המקטלג.

טבלה זו אנו מחזיקים מתוך אילוץ. השימוש ב – Index Service אינו מאפשר ביצוע חיפוש בתוך מסד נתונים, אלא דורש חיפוש בתוך קבצי טקסט, HTML וכדומה. עקב כך אין באפשרותנו לערוך את הרשומות שמחזיק ה Index Service עבור כל מסמך במאגר (רשומות השומרות מספר נתונים עבור כל מסמך), ולהוסיף לרשומה כזו, שדה חדש (כמו קטגוריה למשל). לכן, אנו נאלצים להחזיק טבלה זו ע"מ לשמור את הקטגוריה של כל מסמך ומסמך, כדי שנוכל להשתמש במידע זה מאוחר יותר בעיצוב הפלט עבור כל שאילתה. אופטימלי יותר היה אילו יכולנו לערוך את הרשומות שמחזיק ה Index Service , ולהוסיף את מאפיין הקטגוריה לרשומות אלה. כך כל המידע עבור המסמך (כולל הקטגוריה) היה מוחזר לנו ישירות ע"י ה Index Service ללא צורך בגישה גם לטבלה, דבר אשר היה חוסך את הצורך במבנה נתונים נוסף, וחוסך זמן רב המתבטא בגישות לטבלה ע"מ לשלוף את מאפיין הקטגוריה.

4.1.2.2. טבלת UnClassifiedDocumentsTable

טבלה זו וכן את הספרייה UnClassifiedDocumentsDataBase אנו מחזיקים ע"מ לשמור על רציפות העבודה של המערכת. כלומר, כדי שלא נצטרך לעצור את עבודת המערכת עבור כל קובץ חדש שהורד מהרשת אלא נוכל לבצע כל פרק זמן קבוע עדכון של ה-Database בכמות גדולה יותר של קבצים חדשים מקוטלגים. בצורה זו, נשמור גם על עקביותה של הטבלה הראשית – ClassifiedDocumentsTable.

בטבלה זו תישמר הקטגוריה של כל מסמך בספריית ה- UnClassifiedDocumentsDataBase כפי שתיקבע לו ע"י המקטלג – עד שהמערכת תעביר רשומות אלה לטבלת ClassifiedDocumentsTable, תהליך אשר יתואר בהמשך.

4.2. עיצוב התוכנה

בסעיף זה נתייחס בנפרד לכל אחד משני חלקי המערכת ונתאר את התהליכים המרכזיים שלהם.

4.2.1. מבנה תהליך הקטלוג

תהליך הקטלוג מתרחש "מאחורי הקלעים" של המערכת, ותפקידו להזין את מערכת החיפוש במסמכים חדשים מהרשת ולהעביר מסמכים אלה קטלוג תחת אחת מהקטגוריות המוגדרות. לתהליך זה שני חלקים, הראשון – אתחול המערכת, תהליך המתבצע פעם אחת בלבד. בתהליך זה אנו משתמשים בקבצי ה- training set לאימון תכנת ה- BoosTexter. קבצים אלו יישמרו בספרייה ClassifiedDocumentsDataBase, ויהוו את מאגר המסמכים הבסיסי של המערכת.
התהליך השני הנו תהליך עדכון המערכת המתבצע פעם בשבוע, ותפקידו לקטלג מסמכים חדשים, ולעדכן את מערכת החיפוש במסמכים אלו.

4.2.1.1. אתחול המערכת

חלק זה הנו החלק בו מתבצעת בנייתו של מסד הנתונים ומאגר המסמכים ההתחלתי. תהליך זה מתבצע פעם יחידה בזמן הקמת המערכת.
כפי שהוסבר בסעיף 2.8.2, ע"מ שתכנת ב – BoosTexter תוכל לעבוד כמקטלגת מסמכי טקסט תחת קבוצת קטגוריות מוגדרת, עליה לעבור תהליך של למידה. בתהליך הלמידה היא 'מתאמנת' על קבוצת מסמכים שכבר קוטלגו בצורה ידנית, קבוצה הנקראת - training set – ומסיקה ממנה מסקנות בהן היא משתמשת מאוחר יותר בקביעת קטגוריה למסמך חדש.
השלב הראשון בתהליך הבנייה, הוא העברת קבצי ה- training set לפורמט שמוכר ל- BoosTexter. על שלב זה אחראי ה- DataBaseCreator. תוכנית זו מבצעת תהליך עיבוד מוקדם (preprocessing) על הקבצים. התהליך כולל ניקוי הקבצים מסמני פיסוק מתגי html וממילים סטנדרטיות (כגון at, to וכד') הלקוחות מ- stop list סטנדרטי (ראה נספח ד'), והעברת הטקסט הנקי של הקובץ, לקובץ הקלט של ה- BoosTexter. במקביל, יוצר ה DataBaseCreator רשומה חדשה בטבלת ClassifiedDocumentsTable של שם המסמך וקטלוגו (כאמור, הקטגוריות של המסמכים ב- training set ידועות). בסופו של שלב זה יש בידינו קובץ אחד TrainFile.DATA המייצג את כל קבצי ה train-set בפורמט הנדרש ע"י תכנת ה- BoosTexter , ומסד נתונים המכיל את נתוני המסמכים המקוטלגים בטבלת ClassifiedDocumentsTable.
בשלב הבא, מופעלת תכנת ה- BoosTexter על הקובץ TrainFile.DATA ומבצעת את תהליך הלמידה, שבסופו יתקבל קובץ פלט TrainFile.SHPY בו תשתמש תכנת ה- BoosTexter מאוחר יותר כאשר נשתמש בה לביצוע קלסיפיקציה של מסמך חדש.
את הפעולות שתוארו בסעיף זה מבצע הקובץ TrainProcess.bat.

נתאר את המחלקות המשתתפות בתהליך זה:

א. המחלקה DataBaseConnection :

מחלקה זו אחראית על התקשורת עם מסד הנתונים.

היא אחראית הן לפתיחתו והן לסגירתו של קשר זה, וכל העדכונים והשינויים הנעשים במסד הנתונים נעשים דרכה.

פונקציות המחלקה העיקריות :

- DataBaseConnection : אחראית לאתחול מופע חדש של המחלקה (Constractor) – כולל קריאה ל- Microsoft Access Driver ויצירת קשר עם המסד הנתונים הטבלאי.
- dropTable : אחראית למחיקת טבלה – אם זו קיימת. (אם הטבלה איננה קיימת – אינה עושה דבר).
- createTable : אחראית ליצירת טבלה חדשה – בהנחה שטבלה בעלת שם זהה איננה קיימת.
- createPreparedStatement : אחראית ליצירת אובייקט PreparedStatement (לשליחת שאילתות SQL עם פרמטרים למסד הנתונים).

ב. המחלקה DataBaseCreator :

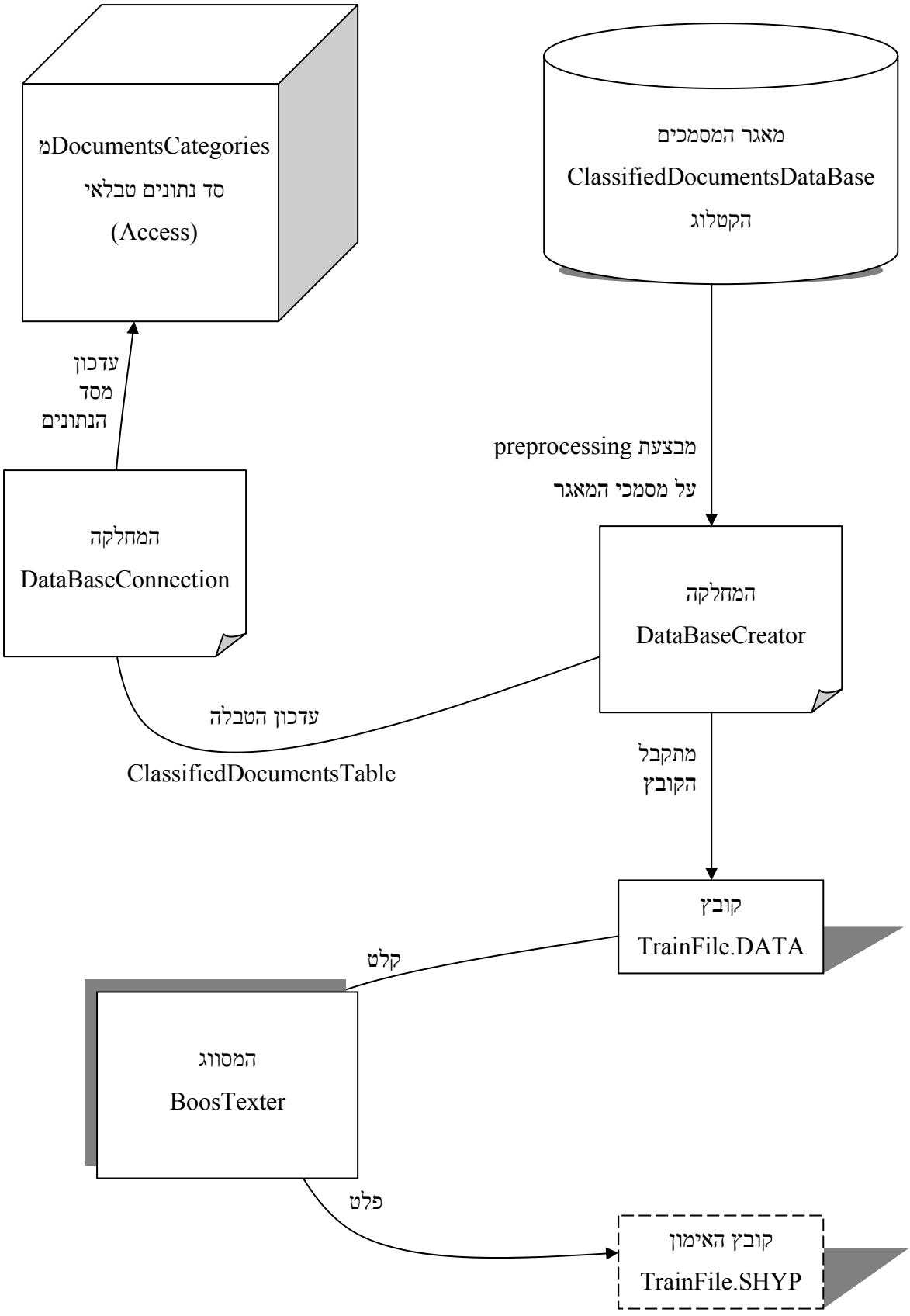
שני תהליכים (תהליך האתחול ותהליך קטלוגם של מסמכים חדשים) עושים שימוש במחלקה זו. נתאר כעת כיצד משמשת המחלקה את תהליך האתחול.

בתהליך האתחול יש צורך לטפל במאגר מסמכים אשר כל מסמך בו קוטלג מראש. לכן פה המחלקה דואגת לעבור על מאגר מסמכים זה וליצור קובץ אחד ארוך, המורכב מכל מסמכי המערכת (עם קטלוגם) לאחר ביצוע סינון קצר. קובץ זה יופנה כקלט, למסווג. כמו כן המחלקה אחראית ליצירת תקשורת עם מסד הנתונים, ליצירת טבלה חדשה ולהכנסת הנתונים - שם המסמך וקטלוגו לטבלת ClassifiedDocumentsTable.

פונקציות המחלקה העיקריות :

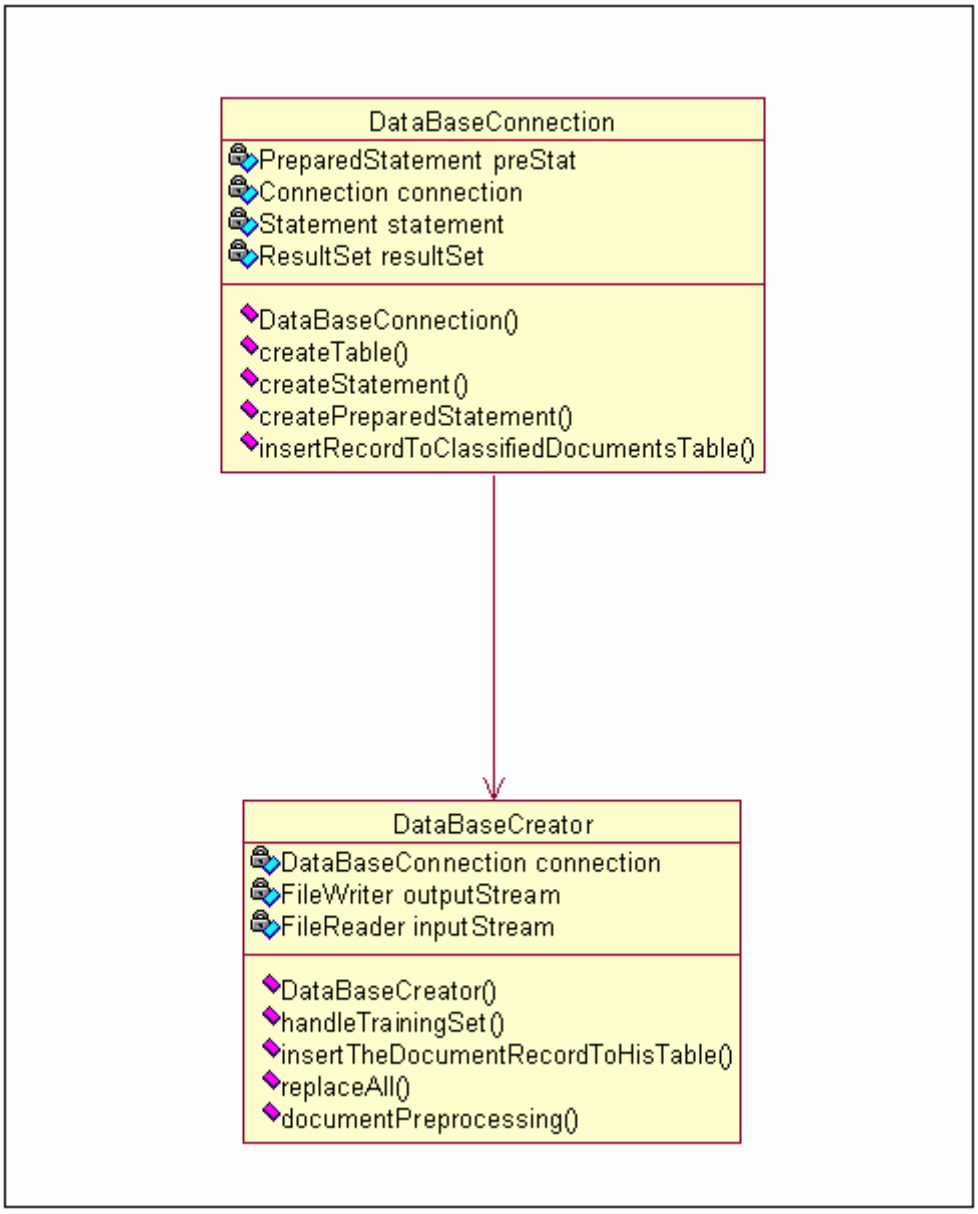
- handleTrainingSet : הפונקציה עוברת על מאגר המסמכים ClassifiedDocumentsDataBase ויוצרת את קובץ הקלט של ה-BoosTexter - TrainFile.DATA. במקביל, יוצרת את טבלת ClassifiedDocumentsTable ומכניסה את שם המסמך ואת הקטגוריה שלו לטבלת ClassifiedDocumentsTable .
- documentPreprocessing : מקבלת מסמך , מנקה אותו מסימני פיסוק לפי דרישות ה-BoosTexter, ומכניסה אותו לקובץ TrainFile.DATA.

איור תהליך האתחול:



איור 6 – איור תהליך האתחול

דיאגרמת המחלקות :



דיאגרמה 1 – דיאגרמת מחלקות של תהליך אתחול המערכת

4.2.1.2. עדכון המערכת

לאחר אתחול המערכת כפי שתואר בסעיף הקודם, המערכת מסוגלת לבצע קטלוג אוטומטי של מסמכים חדשים. התהליך שנתאר בסעיף זה, מתבצע אחת לשבוע ומעדכן את מאגר המסמכים של המערכת, במסמכים חדשים שהורדו מהרשת ועברו קטלוג אוטומטי.

תהליך זה מסתמך על כך, שישנה תוכנת "עכביש" כלשהי (הקיימת בכל מנוע חיפוש ולא מומשה כחלק ממערכת זו) ה"מטיילת" ברשת ומורידה מסמכים חדשים אל ספריית מסמכים זמנית UnClassifiedDocumentsDataBase. לפיכך, נקודת ההתחלה של תהליך זה היא ספריה עם מסמכים חדשים ולא מקוטלגים שהורדו מהרשת. מטרת התהליך היא לקטלג את המסמכים החדשים ורק אז

להעביר אותם מספריה זו למאגר המסמכים הראשי של המערכת, ובדרך זו לאפשר למערכת להשתמש במסמכים אלו.

כל זאת, תוך שמירה על עבודה רציפה ככל הניתן של המערכת, ושימור מסד הנתונים עקבי כל העת.

שלבי התהליך:

ע"מ לאפשר למערכת להשתמש במסמכים החדשים, עלינו לקבוע תחת איזו קטגוריה יקוטלג כל מסמך. אנו נשתמש בתכנת ה- BoosTexter כדי לבצע קלסיפיקציה זו. ולכן הדבר הראשון שייעשה, הוא העברת המסמכים החדשים לפורמט המוכר ל- BoosTexter . דבר זה יעשה בעזרת המחלקה DataBaseCreator שתבצע preprocessing על המסמכים בדומה לתהליך האתחול, ותיבנה את הקובץ TestFile.DATA שהנו קובץ הקלט של ה- BoosTexter. במקביל, תכניס את שם המסמך למסד הנתונים, לטבלת UnClassifiedDocumentsTable.

בשלב הבא נקטלג את המסמכים החדשים. הקטלוג יתבצע ע"י ה- BoosTexter. הקלט שלו בשלב זה יהיה קובץ המסמכים שנוצר כעת - TestFile.DATA, והקובץ שנוצר בתהליך האתחול שתואר לעיל - TrainFile.SHYP (המכיל את חוקי הקלסיפיקציה שהוא יצר). הפלט של ה- BoosTexter, הנו קובץ טקסט פשוט DocumentsClassification.txt המכיל מספר שורות כמספר הקבצים כאשר כל שורה מייצגת קובץ, וכל שורה מכילה דירוג לכל אחת מהקטגוריות ביחס לאותו קובץ. התוכנית UpdateTable עוברת על קובץ זה, ומעדכנת את שדה הקטגוריה ברשומה המתאימה בטבלה UnClassifiedDocumentsTable, בקטגוריה בעלת הדרגה הגבוהה ביותר כפי שקבע ה- BoosTexter לקובץ זה.

כעת משהסתיים התהליך, לכל מסמך נקבעה קטגוריה. אנו מעונינים להעביר מסמכים אלו לשימוש המערכת. נעשה זאת ע"י העברתם לספריית הקטלוג של ה- Index Service , הספרייה ClassifiedDocumentsDataBase שמהווה למעשה את ה- database של המערכת. במקביל, בעזרת התוכנית TableUnion נעביר את שמות המסמכים ואת הקטגוריה שלהם כפי שנשמרו בטבלה UnClassifiedDocumentsTable, לטבלה הראשית ClassifiedDocumentsTable.

הפעולה האחרונה המתבצעת בתהליך זה היא ביצוע אתחול ל- Index Service על מנת שיעדכן את האינדקס שלו במסמכים החדשים.

את הפעולות שתוארו בסעיף זה מבצע הקובץ ClassifyDocuments.bat. קובץ זה יתבצע כמשימה מתוזמנת (Scheduled Task) פעם בשבוע, בשעה מאוחרת בלילה – זמן בו הצפי הוא למספר משתמשים מועט יחסית. עצם הרצת התהליך באצווה, כקובץ batch אינה גוזלת משאבים רבים מהשרת ואינה מאיטה את פעילותו, והעובדה שתהליך זה מתבצע off-line ולא בזמן עיבוד שאילתת החיפוש, מבטיחה עיבוד מהיר של השאילתה והצגת תוצאות תוך זמן קצר מאוד.

נתאר את המחלקות המשתתפות בתהליך זה:

א. המחלקה DataBaseConnection :

כפי שתוארה לעיל, בסעיף 4.2.1.1.

ב. המחלקה DataBaseCreator :

בתהליך העדכון אנו משתמשים בתכנת ה- BoosTexter ע"מ לקטלג את המסמכים. לכן נשתמש במחלקה זו , בדומה לשימוש שעשינו בה בתהליך האתחול, ע"מ לבצע preprocessing על המסמכים החדשים וליצור את קובץ הקלט ל BoosTexter - TestFile.DATA . כמו כן המחלקה אחראית ליצירת תקשורת עם מסד הנתונים, ליצירת טבלה חדשה ולהכנסת שמות המסמכים לטבלת UnClassifiedDocumentsTable.

פונקציות המחלקה העיקריות :

- handleTestingSet : הפונקציה עוברת על מאגר המסמכים UnClassifiedDocumentsDataBase ויוצרת את קובץ הקלט של ה-BoosTexter - TestFile.DATA. במקביל, יוצרת את טבלת UnClassifiedDocumentsTable ומכניסה את שם המסמך לטבלה זו.
- documentPreprocessing : מקבלת מסמך, מנקה אותו מסימני פיסוק לפי דרישות ה-BoosTexter, ומכניסה אותו לקובץ TestFile.DATA.

ג. המחלקה UpdateTable :

מחלקה זו אחראית על קריאת קובץ הפלט של ה- BoosTexter, הקובץ DocumentsClassification.txt , ועדכון שדה הקטגוריה בטבלה UnClassifiedDocumentsTable בהתאם.

פונקציות המחלקה העיקריות :

- readFromFileAndUpdateTable : לאחר שהמסווג קטלג את המסמכים הנמצאים בטבלת UnClassifiedDocumentsTable, הוא מכניס את התוצאות לקובץ פלט. הפונקציה, קוראת את הקובץ, ומעדכנת את הקטגוריה במקום המתאים בטבלת UnClassifiedDocumentsTable.

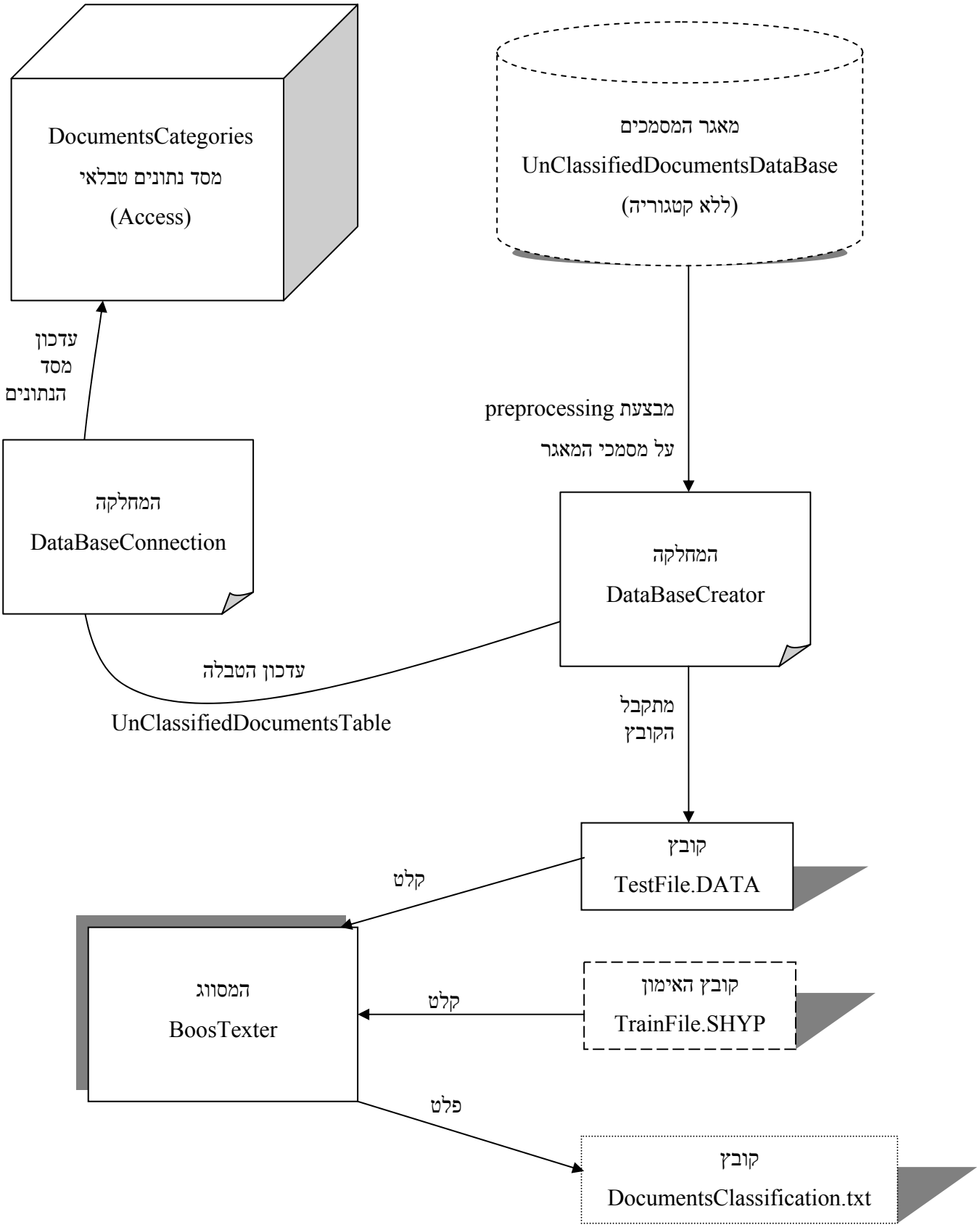
ד. המחלקה TablesUnion :

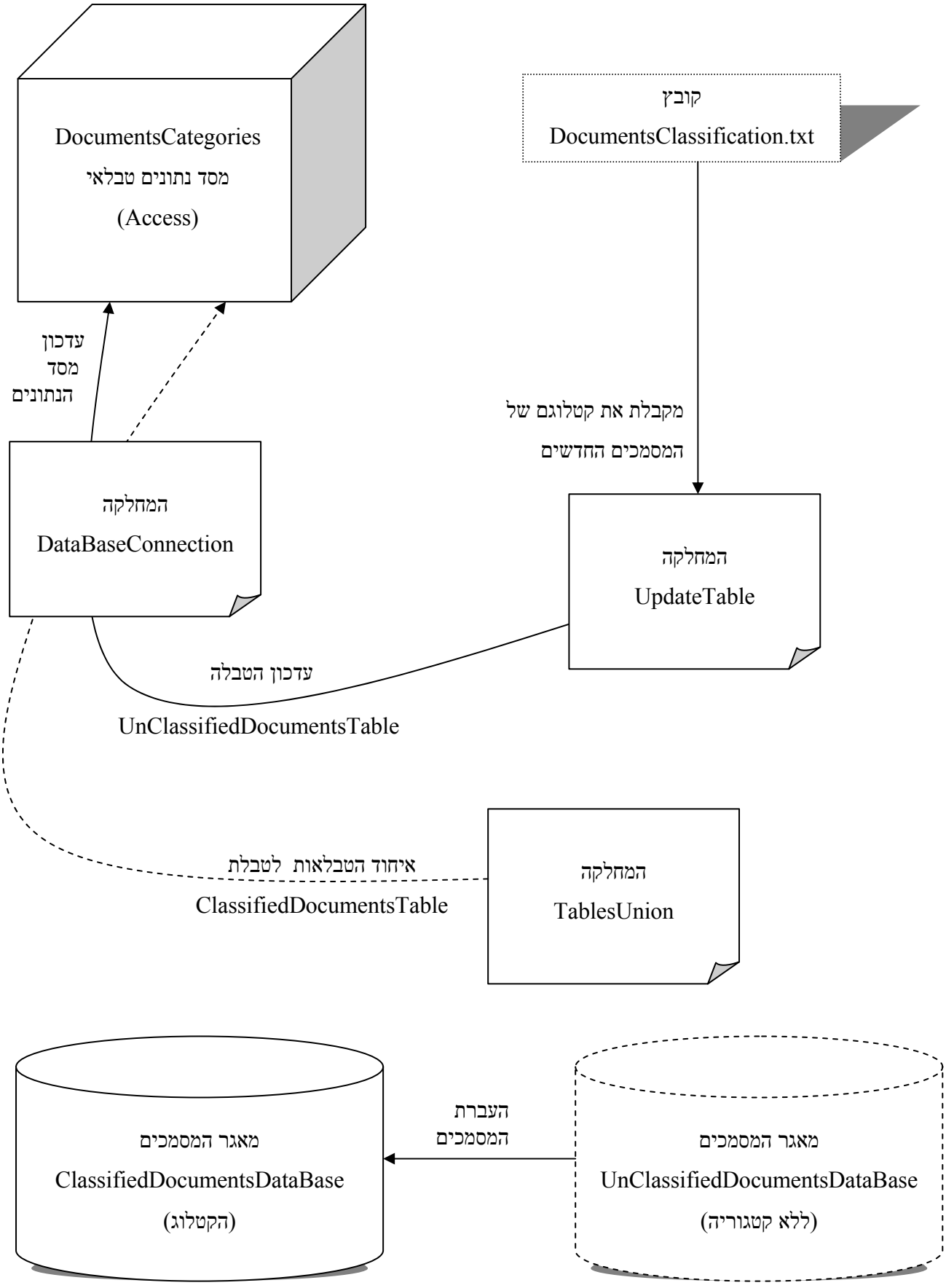
מחלקה זו אחראית על עדכון הטבלה ClassifiedDocumentsTable במסד הנתונים, ברשומות הטבלה UnClassifiedDocumentsTable.

כאשר היא מעבירה רשומה המייצגת מסמך, מהטבלה UnClassifiedDocumentsTable לטבלה ClassifiedDocumentsTable, היא מוסיפה לשמו של המסמך, את ה- path של הספרייה ClassifiedDocumentsDataBase, משום שכפי שצוין לעיל זהו המפתח של טבלה זו.
כאשר מסתיים העדכון, המחלקה מוחקת את הטבלה UnClassifiedDocumentsTable.
פונקציות המחלקה העיקריות :

- main : יוצר תקשורת עם מסד הנתונים, ומאחד בעזרת אובייקט TablesUnion שייצר, את הטבלאות UnClassifiedDocumentsTable ו- ClassifiedDocumentsTable. איחוד הטבלאות מתבצע כך שבסוף התהליך הטבלה ClassifiedDocumentsTable מכילה את נתוני הטבלה UnClassifiedDocumentsTable (הפונקציה אחראית בסוף גם למחיקתה של טבלת UnClassifiedDocumentsTable מממסד הנתונים).

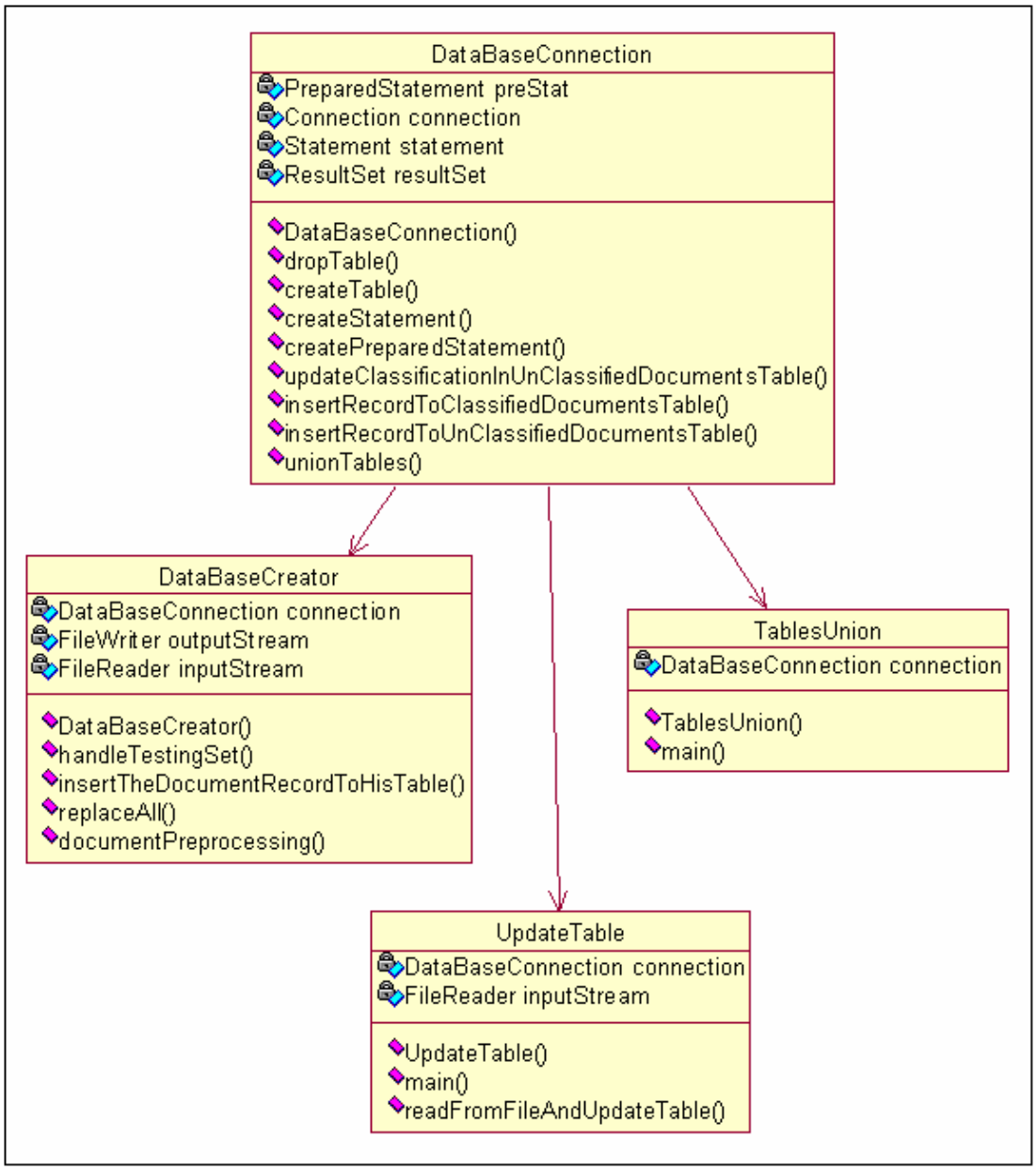
<u>איור תהליך העדכון:</u>





איור 7 – איור תהליך העדכון

דיאגרמת המחלקות :



דיאגרמה 2 – דיאגרמת מחלקות של תהליך עדכון המערכת

4.2.2. ממשק המשתמש

ממשק המשתמש מחולק לממשק קלט וממשק פלט. ממשק הקלט מאפשר למשתמש להגיש שאילתת חיפוש וכן, אם ירצה, לבחור לקבל תוצאות חיפוש רק מקטגוריות מסוימות. ממשק הפלט מציג את תוצאות החיפוש , וכפי שמוסבר לעיל בסעיף 3.2.2.2, ממשק זה הוא ממשק אינטראקטיבי המאפשר למשתמש לבחור ולסנן מסמכים ע"פ הקטגוריה שלהם. בסעיף זה נתאר, כיצד מעובדת שאילתת החיפוש, מה מתבצע בצד של ה- Client, ומה בצד של ה- Server.

4.2.2.1 ממשק הקלט

ממשק הקלט הנו ממשק html פשוט (כמתואר באיור 4) המאפשר למשתמש הכנסת שאילתה, ובחירה בין לראות את כל תוצאות החיפוש, או רק תוצאות חיפוש מקטגוריות נבחרות (כמפורט בסעיף 3.2.2.1). בעזרת פונקציות JavaScript מתבצעת בדיקת תקינות הקלט בצד הלקוח. במידה והקלט תקין הוא מועבר ע"י Post לקובץ resultsOrginaizer.asp.

4.2.2.2 בניית הפלט

הקובץ resultsOrginaizer.asp, מקבל את בקשת החיפוש מצד ה- Client ומעביר אותה לעיבוד ב- Server שם מתבצע עיבוד השאילתה ע"י ה- Index Service , ובניית הפלט. צד ה- Server (שרת IIS של Microsoft) נכתב ב asp.

בעזרת קוד asp מועברת השאילתה אל ה- Index Service. מה- Index Service מוחזרות תוצאות החיפוש בתצורה של Record-Set. ה- Record-Set מכיל כמעט את כל המידע על המסמכים שעונים על תנאי החיפוש, למעט הקטגוריה שלהם, אותה נצטרך לשלוף מטבלת ה- ClassifiedDocumentsTable (זהו אילוץ, כפי שמוסבר בסעיף 4.1.1).

את נתוני המסמכים שחזרו משאילתת החיפוש נשמור בקובץ xml, ע"מ שנוכל לשלוט בצורת התצוגה ובסדר התצוגה של המסמכים , ולאפשר אינטראקציה עם המשתמש. מסמך ה- xml ישמר על ה- session של המשתמש, וכאשר ה- session ייסגר, הוא ימחק. שמירה על ה- session אפקטיבית משתי בחינות- האחת, מחיקת המסמך לאחר סגירת ה- session מונעת פיצוץ של הזיכרון. והשניה, שבכך נמנע ממשתמש אחד להגיע ל- xml המייצג תוצאות חיפוש של משתמש אחר.

במקביל לבניית ה- xml נבנה גם עץ הקטגוריות בעזרת פונקציות JavaScript שהמימוש שלהן נשמר בקובץ treeCode.js.

לאחר בניית קובץ ה- xml והעץ, אנו עוברים להצגת הפלט בצד של ה- Client.

ממשק הפלט יתחלק לשלוש מסגרות:

- מסגרת עליונה- מסגרת דומה לממשק הקלט, מאפשרת הגשת שאילתת חיפוש חדשה.
- מסגרת שמאלית- מסגרת המציגה את עץ הניווט של הקטגוריות, ומאפשרת למשתמש בחירה של מסמכים מכל אחת מהקטגוריות שהופיעו בתוצאות החיפוש.
- מסגרת ימנית מציגה את מסמך ה- xml המייצג את המסמכים שחזרו משאילתת החיפוש, ומאפשרת בחירה של מסמך נבחר.

התצוגה הראשונית של הפלט, מציגה במסגרת השמאלית את עץ הקטגוריות כאשר הוא "סגור" ז"א רואים את קטגורית השורש (הקטגוריה הראשית) ואת תתי הקטגוריות שלה בלבד. במסגרת הימנית מופיעים כל המסמכים שחזרו מכל הקטגוריות. תצוגת מסמך ה- xml נעשית ע"פ מסמך ה- xsl(Extensible Stylesheet Language) הנשמר בשרת, ומשותף לכל המשתמשים.

בשלב הבא מתבצעת אינטראקציה עם המשתמש. המשתמש יכול לצפות באחד המסמכים המוצגים לו, או למקד יותר את תוצאות החיפוש ולסנן מסמכים ע"פ הקטגוריה. דבר זה נעשה בעזרת עץ הניווט, המאפשר למשתמש לבחור קטגוריה מכל אחת מרמות העץ. כאשר המשתמש בוחר בקטגוריה מסוימת, מעודכן פרמטר בקובץ ה- xml. ע"פ פרמטר זה קובע קובץ ה- xsl כיצד להציג את מסמך ה-xml. מסמך xsl זה מבצע בעצם מיון של מסמך ה-xml ושולף מתוכו רק את הרשומות מהקטגוריה הנבחרת. כעת יוצגו במסגרת הימנית רק נתוני מסמכים אלה. כך יכול המשתמש לבחור בעץ הקטגוריות כרצונו ולבצע סינון מסמכים המאפשר לו למקד את החיפוש שלו.

המחלקות והפונקציות הבונות את העץ כתובות ב- JavaScript.

נתאר את המחלקות המשתתפות בתהליך זה:

א. המחלקה Folder:

מחלקה זו מתארת קודקוד פנימי בעץ. קודקוד פנימי בעץ יכול לייצג את קטגוריית השורש, או אחת מתתי הקטגוריות שלה. "הבנים" (בעץ) של כל Folder, יכולים להיות גם הם מסוג Folder, במקרה שמדובר בקודקוד המייצג קטגוריה שיש לה תת קטגוריה. או, אם מדובר בתת קטגוריה ברמה הנמוכה היותר, ה-Folder שמייצג אותה, יכיל אובייקטים מסוג Item, המייצגים מסמכים מאותה קטגוריה.

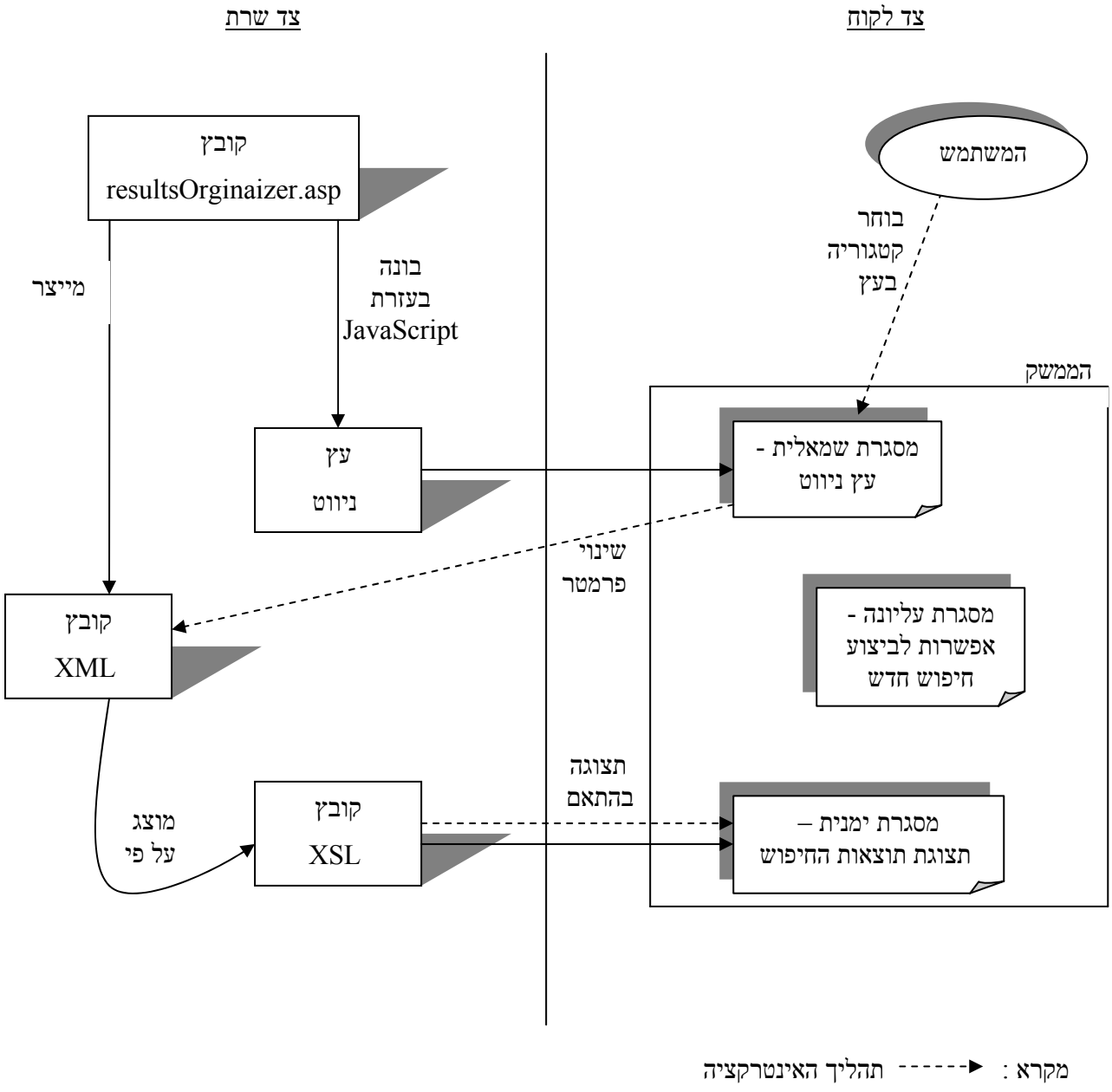
ב. המחלקה Item:

מחלקה זו מייצגת למעשה את עלי העץ, ז"א את המסמכים עצמם, או ליתר דיוק – לינקים אל המסמכים.

הפונקציות העיקריות :

- createFolder(name, filename) : הפונקציה, מקבלת שם של Folder אותו רוצים ליצור, ו-reference לקובץ שאנו רוצים שיפתח בעת לחיצה. הפונקציה יוצרת ומחזירה Folder.
- insertFolderToFolder(parentFolder, childFolder) : הפונקציה מקבלת איזשהו קודקוד בעץ, parentFolder, ומכניסה את childFolder כבן שלו בעץ.
- createLink(target, title, url): יוצרת קישור (Link) למסמך. target – היכן (באיזה חלון) לפתוח את המסמך. title – הכותרת של המסמך אשר תופיע על הקישור. url - מיקום המסמך.
- insertDocument(Folder, createLink(…)): מכניסה קישור למסמך לתוך התיקייה הנתונה.

תאור הקשר Client-Server :



איור 8 - תאור הקשר Client - Server

5. דיון

בשנים האחרונות, משימת עיצוב ממשק משתמש למערכות מידע, תופסת מקום מרכזי בתהליך תכנון מערכת חיפוש ופיתוחה. זאת, כתוצאה מן הגידול המתמיד בכמות המידע המצוי באינטרנט.[11]. רוב מנועי החיפוש, אשר מטפלים בעיקר בטקסטים, נאלצים להציג כמות גדולה של מסמכים במענה לפעולת חיפוש שמבצע המשתמש במנוע החיפוש. תוצאות החיפוש, מוצגות לרוב כרשימה ארוכה של מסמכים, אשר המידע הנתון עבור כל מסמך הוא ע"פ רוב כותרתו, כתובתו ומשפט קצר מגוף המסמך המכיל את ערך החיפוש. המתח הקיים - בין אילוץ מנוע החיפוש להציג כמות מסמכים רבה לבין סבלנותו המוגבלת של המשתמש, הנוטה להתבונן בלא יותר מעשרים המסמכים הראשונים ברשימה - מחייב טיפול מיוחד בממשק המשתמש במערכות אחזור מידע.

"CBI Searcher" (Categories Based Interface Searcher) מנסה לפתור סתירה זו באמצעות ממשק תוצאות נוח, המאפשר ניווט מהיר ופשוט ברשימת התוצאות הארוכה המתקבלת. ובעזרת מתן אפשרות בחירה מהירה של קבוצת מסמכים מסוימת.

5.1. יתרונות וחסרונות המערכת- הצגה השוואתית

טיוב איכות השירות הניתן ע"י מנוע חיפוש למשתמשיו, איננו ניתן לביצוע רק באמצעות עיבוד שאילתה מהיר יותר או מתן מענה מקיף יותר. קיומו של ממשק נח למשתמש הנו מרכיב קריטי באיכותו של כל מנוע חיפוש. ה-"CBI Searcher" מתמקד בסוגיה זו ובכך נותן מענה טוב יותר למשתמשיו.

5.1.1. יתרונות ממשק ה-"CBI Searcher"

1. **יכולת שליטה של המשתמש בתוצאות המתקבלות לידיו**

ה-"CBI Searcher" מכיל "עץ ניווט" אשר נבנה באופן דינמי עבור כל שאילתה. רכיב זה מאפשר למעשה קיומה של אינטראקציה מתמשכת בין המערכת למשתמש. אינטראקציה זו מאפשרת שליטה של המשתמש בתוצאות החיפוש, הבאה לידי ביטוי ב:

יכולת סינון מסמכים רבים אשר אינם רלוונטיים כלל לנושא בו מתעניין המשתמש.

אפשרות בחירה מהירה של קבוצת מסמכים הנמנית על קטגוריה רלוונטית לערך המבוקש. באמצעותה מתמקד החיפוש.

2. **הגדלת הסיכוי להשגת המבוקש תוך פרק זמן קצר יחסית**

ה-"CBI Searcher" מאפשר למשתמש להתמודד באופן יעיל ומתקדם עם רשימת מסמכים ארוכה המתקבלת לידיו, ומגדיל את סיכוייו להגיע אל המבוקש לו. השימוש בו, חוסך מן המשתמש קריאה ארוכה ומייגעת של רשימת מסמכים אינסופית אשר חלקים ניכרים בה אינם רלוונטיים עבורו. תהליך זה גורם לעיתים קרובות לתסכול רב מצדו של המשתמש ומונע ממנו לאתר את מבוקשו.

3. **<u>יכולת סינון מתקדם של החיפוש עוד בטרם קבלת התוצאות</u>**

ה-"CBI Searcher" מאפשר לבחור, כבר בשלב הגדרת הערך המבוקש, את הקטגוריות בהן יתמקד החיפוש. מהלך זה, מקנה למשתמש את היכולת לצמצם מראש את מספר המסמכים המתקבלים לידיו ולמקד מבעוד מועד את התוצאות המתקבלות.

4. **<u>שימוש נוח במערכת</u>**

נוחות ופשטות השימוש ב-"CBI Searcher", המכיל ממשק אינטראקטיבי, הופכת את חווית/משימת החיפוש לנעימה יותר. נציין כי השימוש בממשק, אינו מחייב שינוי בהרגלי החיפוש. נתוני המסמכים עצמם מוצגים באותו אופן בו מציגים אותם מנועי החיפוש המוכרים. הממשק ברור וקל לשימוש ואינו דורש ידע מוקדם או לימוד.

5.1.2. <u>חסרונות ממשק ה-"CBI Searcher"</u>

1. **<u>הצורך בקטלוג מוקדם של כל מסמך</u>**

טרם צירופו של כל מסמך חדש אל מאגר המסמכים של המערכת, יש חובה לקבוע את הקטגוריה אליה ישתייך. תהליך קטגוריזציה זה, מחויב להתבצע בכל מסמך המגיע מהרשת אל המערכת, לפני שניתן לעשות בו שימוש במערכת. חשוב לציין עם זאת, כי כל מנוע חיפוש מבצע עיבוד מוקדם על המסמכים בבסיס הנתונים שלו.

2. **<u>קטלוג סובייקטיבי העלול לחבל ביכולתו של המשתמש לאתר את מבוקשו</u>**

ה-"CBI Searcher" מחייב חלוקה של כלל המסמכים במאגר לקטגוריות. כל מסמך באשר הוא, חייב להיות משויך לקטגוריה נושאית מסוימת, ולאחת בלבד. קטלוג הנו תהליך סובייקטיבי, אשר על כן, קיימת סבירות להיווצרותו של פער בין הקטלוג שעורכת המערכת לבין זה הנערך ע"י כל משתמש. בעוד המערכת תסווג מסמך X לתחום נושאי מסוים, אפשר שהמשתמש יסבור כי ימצא את מבוקשו (X) דווקא תחת קטגוריה אחרת. פער זה עלול להקשות על המשתמש לאתר את מבוקשו. עם זאת, מבחינה מקצועית של אנשי מידע וספרנים, קטלוג הוא כמעט כמו מתמטיקה מבחינה זו שהוא די מוחלט ויש טבלאות מקובלות עולמיות לכל תחום הידע.

3. **<u>העדר דיוק בקטלוג על בסיס סמנטי</u>**

ב-"CBI Searcher" מתקיים תהליך קיטלוג אוטומטי. תהליך זה, אינו משלב חשיבה אנושית אלא מסתמך על סמנטיקה בלבד (זיהוי מילים רלוונטיות בגוף הטקסט). כתוצאה מכך עלול להיווצר עיוות באיכות הקיטלוג (ייתכן ומסמך מסוים יקוטלג ע"פ המילים המרכיבות את הטקסט לקטגוריה X כאשר בפועל יש לסווגו תחת קטגוריה Y). עיוות שכזה יכול להשפיע על יכולתה של המערכת להביא את המשתמש אל המידע המבוקש לו.

4. **<u>העדר דיוק בקטלוג מסמך המכיל טקסט קצר</u>**

היות והקיטלוג ב-"CBI Searcher" הנו על בסיס סמנטי, קיים קושי לדייק בקיטלוגם של מסמכים המכילים טקסט המורכב משורות בודדות בלבד. לפיכך מסמכים קצרים עלולים להתמקם תחת קטגוריה בלתי רלוונטית.

5.2. <u>שינויים ושיפורים אפשריים</u>

ישנן מספר אפשרויות לשיפור והרחבת המערכת, הן בתחום מנוע החיפוש ויכולותיו והן בתחום הקטלוג האוטומטי. נפרט להלן את חלקן. לא נתייחס לשיפורים אשר ניתן לערוך בחומרה או במשך החיפוש.

1. **<u>שימוש במנוע חיפוש ולא ב- Index Service</u>**

שימוש במנוע חיפוש במקום ב- Index Service, עשוי לשפר את ביצועיה של המערכת הן ביכולות החיפוש, הן במשך החיפוש והן במידע הנשמר ע"י מנוע החיפוש עבור כל מסמך העונה לשאילתת המשתמש. כמו כן, השימוש במנוע חיפוש, יבטל את הצורך בטבלת עזר לשמירת הקטגוריה, דבר אשר יקצר את זמן הריצה בתהליך עיבוד השאילתה.

2. **<u>שיפור ביצועי מקטלג הטקסט האוטומטי</u>**

שיפור בתחום זה ניתן לבצע בשני אופנים:

א. שיפור ביצועי ה- BoosTexter. שיפור זה ניתן לבצע ע"י הגדלה משמעותית של מספר המסמכים ההתחלתי – ב- Training Set, דבר אשר עשוי לשפר את תהליך הלמידה של המקטלג ובכך לשפר את ביצועיו.

ב. השקעה בפיתוח כלי חדש לקטלוג אוטומטי של טקסט, בעל ביצועים טובים יותר.

3. **<u>הרחבת עץ הניווט לעץ שיכלול את העולם כולו</u>**

עץ הקטגוריות בו אנו עורכים שימוש במערכת זו, מייצג רק חלק קטן מעולם הנושאים בהם עוסקים המסמכים ברשת. על מנת להקיף את עולם הנושאים יש להרחיב את עץ הקטגוריות כך שיכלול קטגוריות נוספות המייצגות את הנושאים החסרים.

6. אפשרויות שיווק

ניתן לשווק את ה-"CBI Searcher" בשני אופנים:

1. כרעיון אפשרי למימוש

ניתן לשווק את המערכת כרעיון, אותו יוכלו המפתחים לשפר, לקדם ולהתאים למנוע חיפוש קיים שברשותם. ייטב בשלב זה, לשווק את ה-"CBI Searcher" כרעיון בלבד ולא כמערכת שלמה. זאת, עקב המגבלות השונות המתקיימות במערכת. ביניהן, שימוש ב- Index Service כמנוע חיפוש, אשר מחייב פיתוח בסביבת Microsoft. או העדרן של המכונות והטכנולוגיות הקיימות במנועי החיפוש המקובלים. מגבלות אלו עלולות להקשות על שיווקה של המערכת, לאחד ממנועי החיפוש הידועים, כמערכת מובנית אשר תחליף או תשתלב במערכות הקיימות של מנוע החיפוש. עם זאת, ניתן להציע למנועים אלה את דרך תצוגת התוצאות בממשק אינטראקטיבי המוצגת במערכת זו, כדרך לשיפור השירות למשתמשי מנוע החיפוש.

2. כמערכת שניתן לעשות בה שימוש כמו שהיא

אפשרות נוספת, היא לשווק את המערכת כמו שהיא, כמנוע חיפוש בפני עצמו. ישנם מאגרי מידע ברשתות אינטרא-נט או מאגרי מידע מקוונים, המאחסנים מסמכים העוסקים בתחום מסוים. כך למשל, בתחום הרפואה קיים מאגר ה –Med line ובתחום העיתונות קיים ארכיון המסמכים של רויטרס. כאשר אנו עוסקים במאגר של מסמכים העוסקים בתחום מסוים, קל יותר ליצור חלוקה של תחום זה לתתי תחומים (במאגרים שכאלה בד"כ כבר קיימת חלוקה שכזו), בצורה היררכית, וליצור ממנו עץ קטגוריות הנדרש לפעולת המערכת. חלוקה כזו של תחום נתון, מאפשרת קטלוג טקסט אוטומטי אמין יותר, דבר המאפשר שימוש נכון יותר ומתאים יותר במערכת זו כמנוע חיפוש בפני עצמו.

יש לציין, כי לא ניתן מבחינה חוקית לשווק את המערכת בתצורתה הנוכחית. זאת, עקב השימוש ב- BoosTexter , אשר מותר בשלב זה, לצורכי לימוד בלבד. בכדי להפשיר המערכת לשיווק מסחרי, יש לבדוק מהם התנאים שמציבה חברת AT&T לכך.

7. סיכום

הגידול בהיקף מאגרי המידע בעולם, אשר בא לידי ביטוי בולט ברשת האינטרנט, מחייב התייחסות מיוחדת בכל הקשור לממשק משתמש. זאת, ע"מ להקל על המשתמש במציאת המידע המבוקש. כיום, רוב מנועי החיפוש ברשת דומים בדרך הצגת המידע שלהם למשתמש. חיפוש ממוצע אשר עורך המשתמש, נותן בידיו רשימה סדרתית ארוכה המונה אלפי מסמכים העונים לתנאי החיפוש. מחקרים מראים כי משתמש ממוצע נוטה להתבונן בלא יותר מעשרים המסמכים הראשונים המופיעים ברשימה. מכאן ניתן להסיק כי, צורת הצגה זו אינה מאפשרת למשתמש לבחור באופן מהיר ויעיל את המידע הרלוונטי לגביו.

פרוייקט זה, "עידון תהליכי חיפוש במאגרי מידע והצגתם למשתמש", דן באופן הצגת תוצאות חיפוש במערכות מידע ומציע דרך נוחה ויעילה יותר, להצגת התוצאות. זאת, באמצעות בנייה של ממשק אינטראקטיבי הנתמך ע"י עץ ניווט, המאפשר למשתמש לסנן מסמכים רבים אשר אינם בתחום עניינו, ולהתמקד במסמכים השייכים לקטגוריות הרצויות לו בלבד. כתוצאה מכך חל טיוב בתפוקות המשתמש אשר בא לידי ביטוי, הן ביכולתו למצוא את המסמך המבוקש לו, והן בקיצור הזמן אותו הוא משקיע במציאתו.

מימושה של המערכת המתוארת לעיל מתבסס על קטגוריזציה של מסמכים. (מהלך מעין זה קיים במערכת ה- LCC&K (Line in Context, Categories, & Keywords) אשר נבחנה באוניברסיטה העברית בירושלים) קטלוגו של כל מסמך בבסיס הנתונים, נקבע במערכת באופן אוטומטי. הקטלוג נעשה ע"י כלי לקטלוג אוטומטי של מסמכי טקסט – BoosTexter. קטלוג כל מסמך תחת קטגוריה רלוונטית מחלק למעשה את כלל המסמכים הקיימים בבסיס הנתונים לקבוצות. כך מתאפשר לכל משתמש לחפש את המסמך המבוקש לו באמצעות חיפוש ממוקד בקטגוריות הרלוונטיות עבורו.

המערכת שבנינו מהווה בסיס לרעיון, אותו ניתן ליישם על מנועי חיפוש קיימים.

עיצוב ומימוש המערכת דרש מאתנו התמודדות עם נושאים תיאורטיים ומעשיים רבים, לימוד שפות וטכנולוגיות חדשות. במסגרת זאת, הובלנו תהליך שתחילתו בהכרת הבעיה ולימוד התחום עליו היא נמנית, וסיומו במימוש מערכת המספקת פתרון אופציונלי לבעיה זו.

8. ביבליוגרפיה

[1] S. H. Lin, C. S. Shih, M. C. Chen, J. M. Ho, M. T. Kao, and Y. M. Huang. "Extracting Classification Knowledge of Internet Documents: A semantics Approach". 1998. URL: http://kp05.iis.sinica.edu.tw/shlin/paper/SIGIR98.pdf

[2] O. Drori. "Improving Display of Search Results in Information Retrieval Systems - User's Study". 2001. URL: http://shum.huji.ac.il/~offerd/papers/drori072001.pdf

[3] O. Drori. "User Interface as Information Filtering". 1999. URL: http://shum.huji.ac.il/~offerd/papers/drori051999h4.pdf

[4] C. J. Van Rijsbergen. Information Retrieval. Butterworths, London, Jan 1979.

[5] C. Apte, F. Damerau, and S. M. Weiss. "Automated Learning of Decision Rules for Text Categorization". 1994. URL:
http://www.research.ibm.com/dar/papers/pdf/tois94_with_cover.pdf

[6] S. Brin and L. Page. "The Anatomy of a Large-Scale Hypertextual Web Search Engine". 1998. URL : //www7.scu.edu.au/programme/fullpapers/1921/com1921.htm

[7] S. Dumais, J. Platt, D. Heckerman and M. Sahami. "Inductive Learning Algorithms and Representations for Text Categorization". 1998 .URL: http://robotics.stanford.edu/users/sahami/papers-dir/cikm98.pdf

[8] Yihua Liao and V. Rao Vemuri, "Using Text Categorization Techniques for Intrusion Detection". 2001. URL: http://www.cs.rpi.edu/~brancj/publications/Liao-Vemuri.pdf

[9] F. Sebastiani . "Machine Learning in Automated Text Categorization". 2002. URL:
http://webster.cs.uga.edu/~miller/SemWeb/Presentation/ACTfiles/ACMCS02.pdf

[10] R. Schapire and Y. Singer, “BoosTexter: A Boosting –based System for Text Categorization”.2000. URL: http://www.cs.huji.ac.il/~singer/papers/boostexter.ps.gz

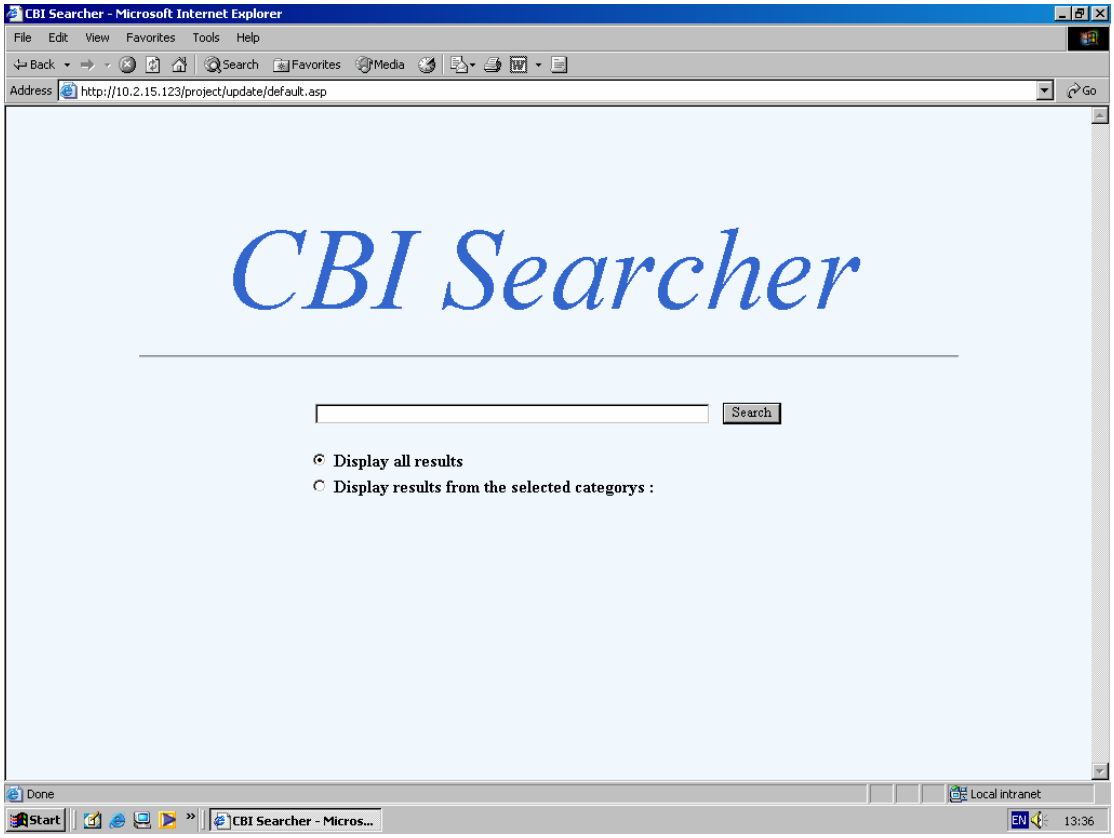
[11] O. Drori. "Using Text Elements by Context to Display Search Results in Information Retrieval Systems". 2000. URL:
http://shum.huji.ac.il/~offerd/papers/utecdsr/drori052000.htm

9. נספחים

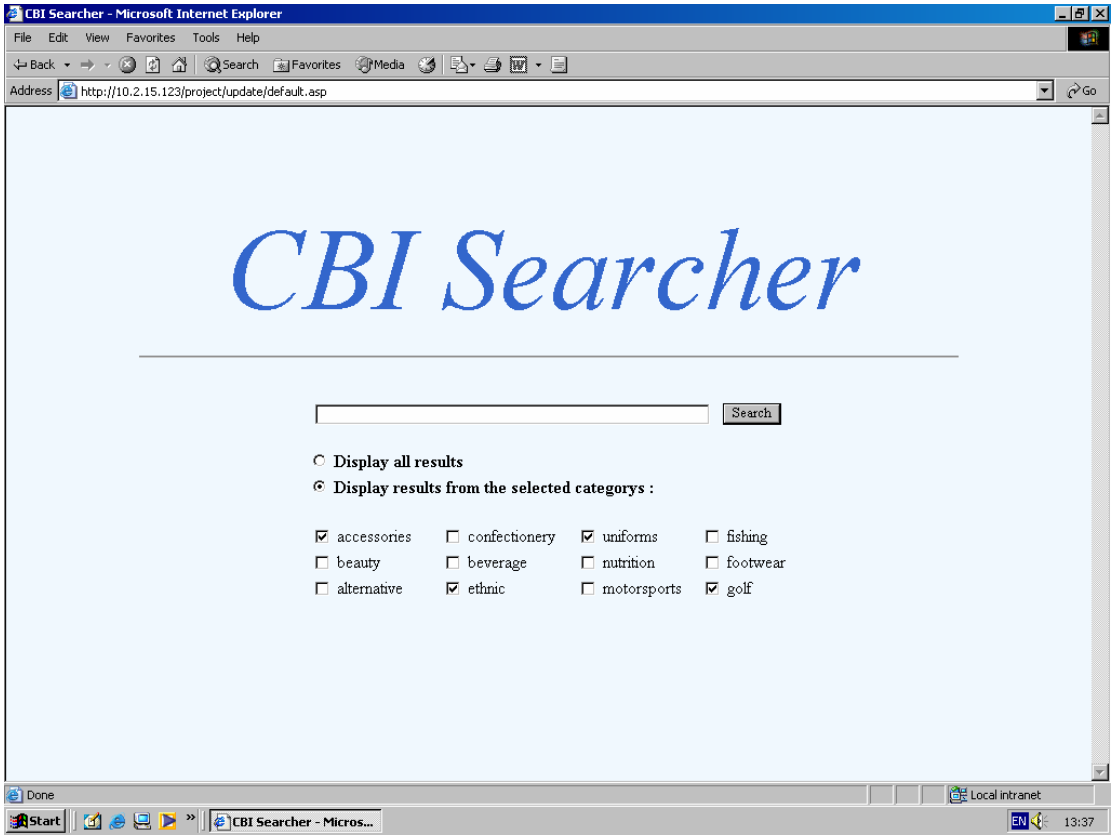
נספח א' - רשימת איורים

נספח ב' - רשימת דיאגרמות

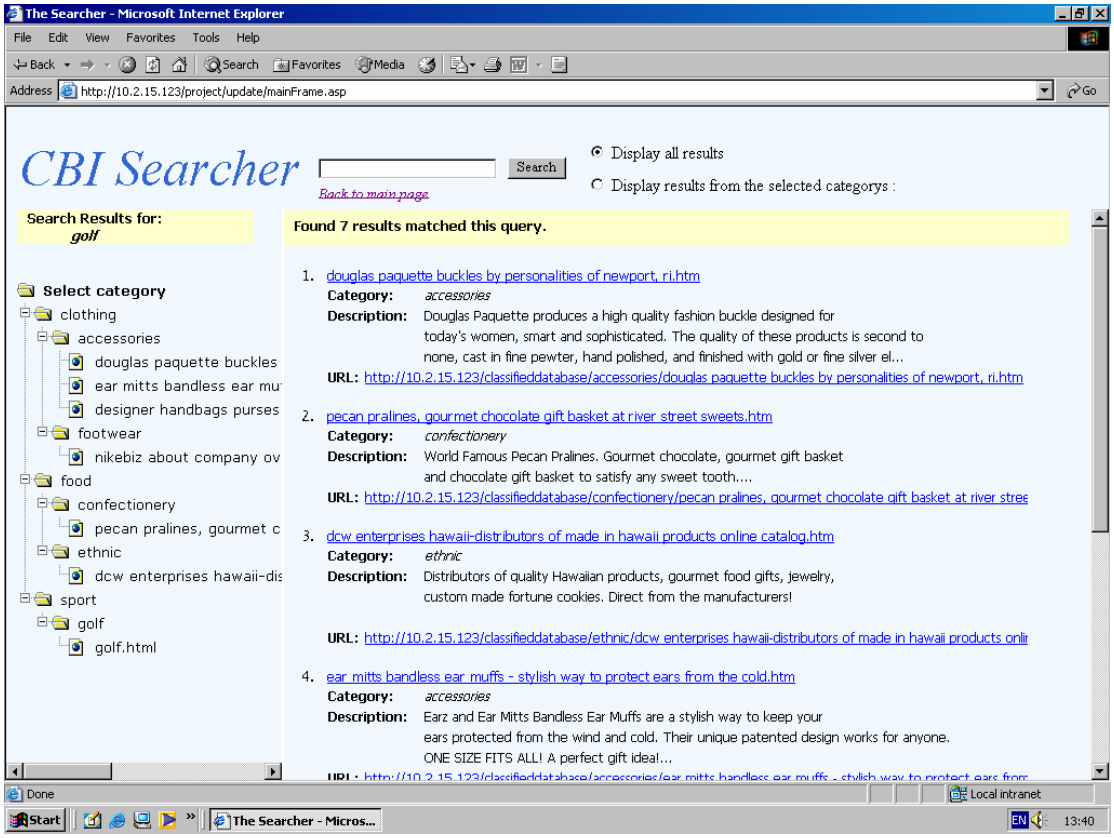
נספח ג' - מסכים עיקריים של המערכת



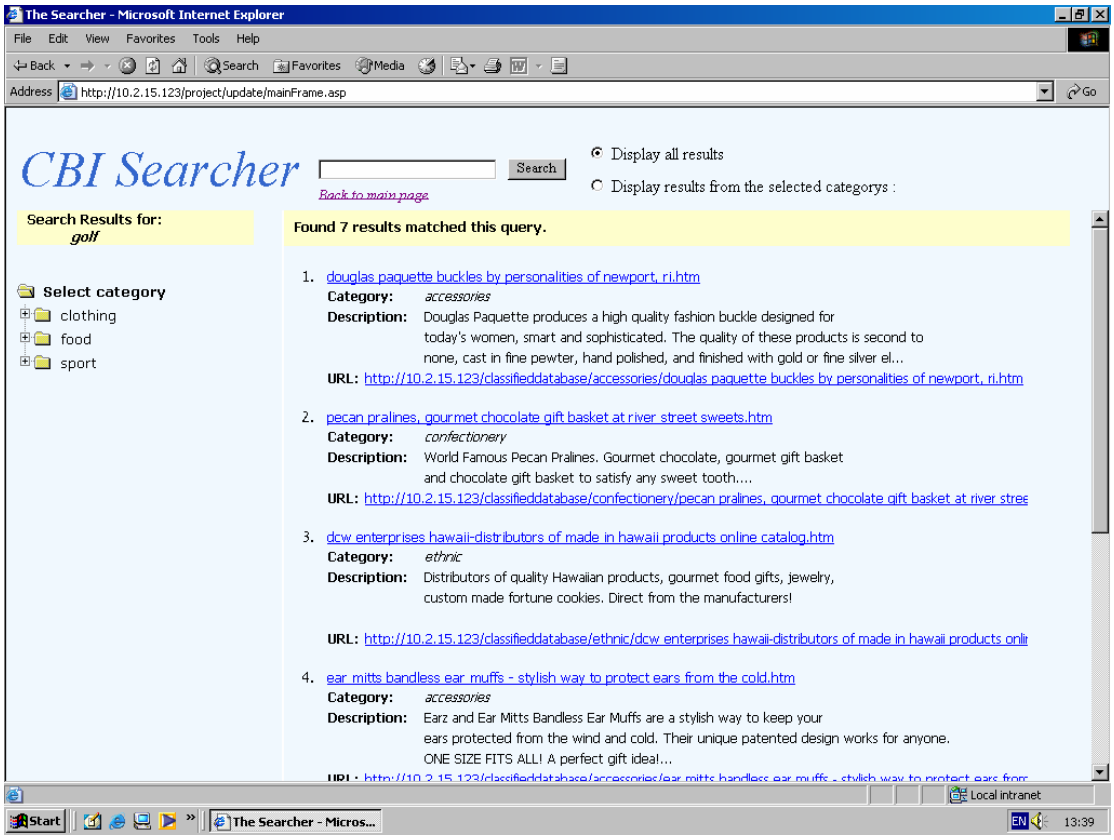
מסך 1 - מסך ראשי



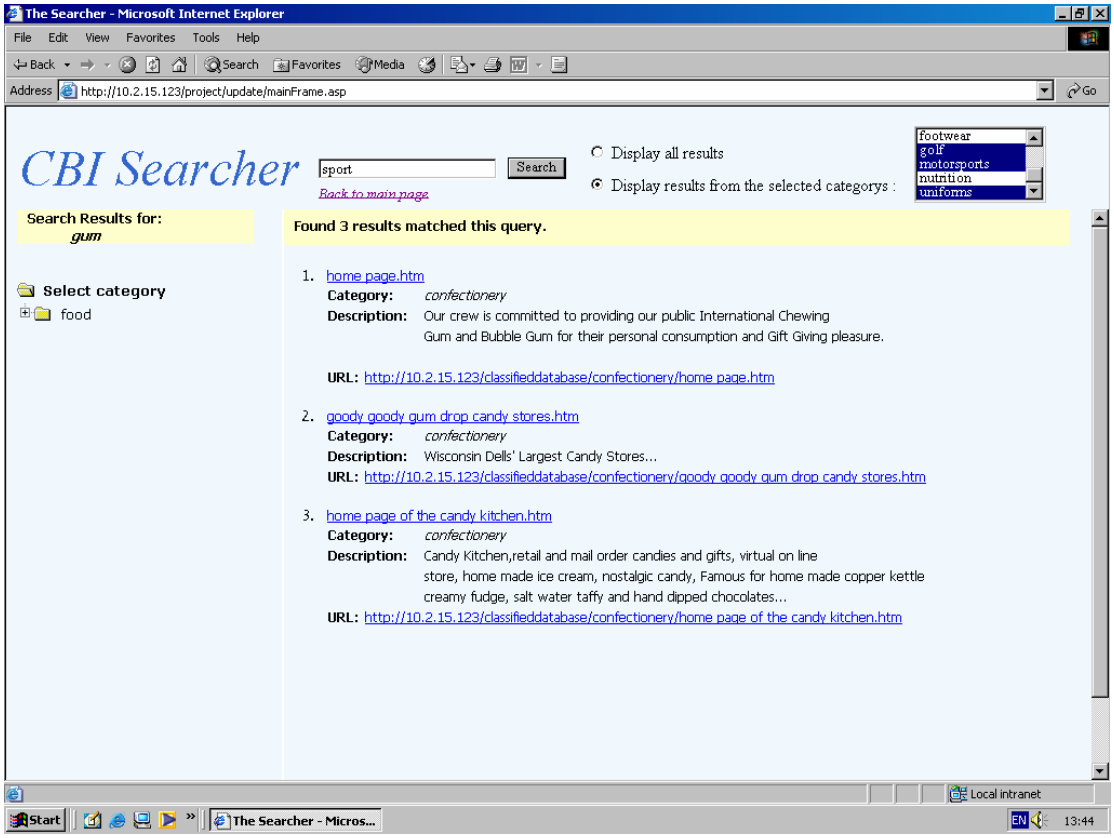
מסך 2 - מסך ראשי : אפשרות בחירת קטגוריות



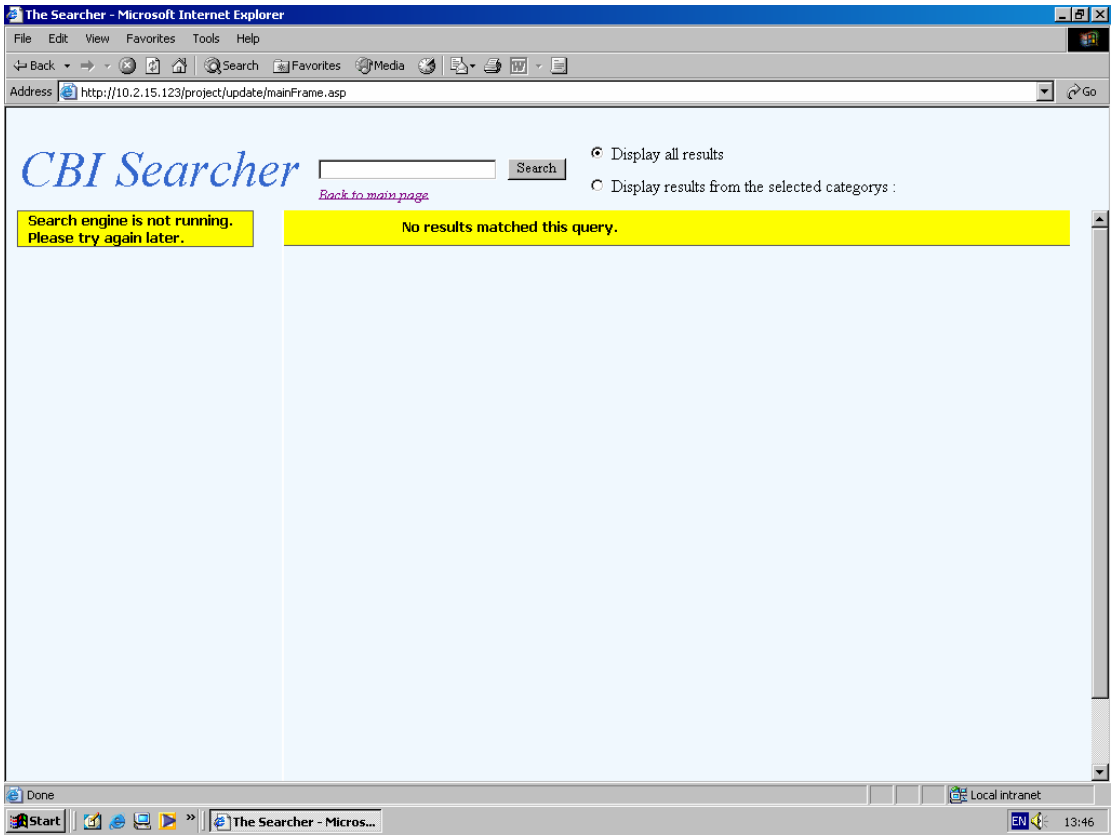
מסך 3 – מסך בו עץ הניווט פתוח



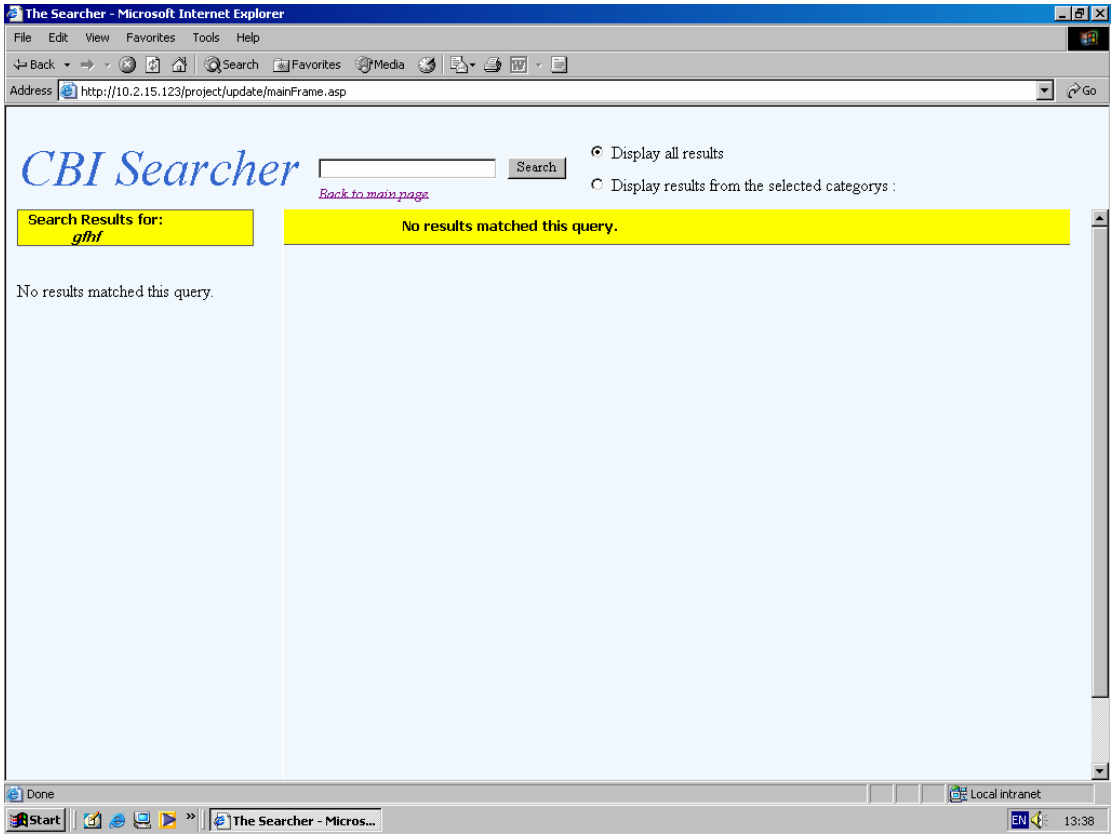
מסך 4 - מסך בו עץ הניווט סגור



מסך 5 - בחירת קטגוריות



מסך 6 - מנוע החיפוש אינו עובד



מסך 7 - אין תוצאות המתאימות לערך המבוקש

Stop list - נספח ד'

as	an	a	I
his	he	by	at
us	thou	or	me
amidst	amid	against	who
anybody	and	amongst	among
circa	beside	because	anyone
everyone	everybody	during	despite
hers	her	from	for
hisself	himself	him	herself
it	into	if	idem
nor	myself	itself	its
our	onto	oneself	of
she	per	ourselves	ourself
the	that	than	since
themselves	them	theirs	thee
thyself	this	thine	they
towards	toward	tother	to
versus	upon	until	unless
whatall	what	we	via
whichsoever	whichever	which	whereas
whomso	whomever	whom	whoever
with	whosoever	whose	whomsoever
you-all	you	ye	without
aboard	yourselves	yourself	yours
after	across	above	about
although	alongside	along	all
anything	any	anti	another
bar	aught	astride	around
below	behind	before	barring
beyond	between	besides	beneath
considering	concerning	but	both
enough	either	each	down
few	excluding	excepting	except
in	ilk	following	fewer
many	like	inside	including
most	more	minus	mine
nobody	neither	near	naught
off	notwithstanding	nothing	none
otherwise	other	opposite	on
past	own	over	outside
round	regarding	plus	pending
so	several	self	save
something	someone	somebody	some
sundry	suchlike	such	somewhat
throughout	through	though	there
underneath	under	twain	till
vis-a-vis	various	up	unlike
wherewith	when	whatsoever	whatever
worth	within	while	wherewithal
is	yonder	yon	yet

Abstract

The World-Wide Web (WWW) is a vast repository of information, much of which is valuable but very often hidden to the user. To find information on the Web, a number of search engines have been developed which gather material, each by its own method, indexing words in the database and searching them. Most search engines are similar in the way they display their information for the user. A list of document fulfilling the search condition is displayed in a serial fashion while carrying out a partial attempt in ranking the list. This list often contains hundreds and sometimes even thousands of documents. Some research found that the average user looks at only the first 10-20 items in the search results list. In this case, the user has limited options to continue the search, and aid is needed to help the user reach the desired document.

The goal of this project is planning and writing of a search system that is based on an existing search engine, which will exhibit a possible solution to this problem. Using an interactive user interface, the system will be able to filter out the many documents that are irrelevant to the user. The presentation of the search question and the search in the data bank , will be done as in any search engine. The search results will be grouped according to their categories. The category will be defined with a text classifier, that will work off-line on the document bank, and will decide for each document under which category, of the earlier decided groups of categories, the document will be catalogued. After the documents in the search result have been grouped according to their category, the categories will be presented to the user as a hierarchic tree. The user will be allowed at every stage, to decide if to see all the documents grouped under this category (if there any), or to leave the tree for a sub-category of this category (if there is one), and see only the documents catalogued here. In this way, the interface will allow the user to focus on the documents of the subject relevant to him, and will filter out the many documents that fulfill the search question but are irrelevant to the area the searcher is looking for.

Credits

We would like to thank Dr. Offer Drori for advising us in the project.

THE HADASSAH ACADEMIC COLLEGE
DEPAETMERNT OF COMPUTER SCIENCE

FINAL REPORT

Refinement procedures of searching in data banks and presenting them to the users

SUBMITTED BY: Ziv Slater

Rachel Ben-Ezra

ADVISOR: Dr Offer Drori

JERUSALEM, October 2003

אינדקס לכרכים א׳ עד י׳ עפ״י מחברים

אינדקס לפי שם מחבר (כרך א' עד כרך י' חוברת 2)

שם המאמר	שם המחבר	כרך	גליון מס.	תאריך
XML והשלכותיו על בסיסי נתונים ואחזור טקסט	אבגי ראובן	ו'	2	6/2000
מנוע האחזור Inter Text - גרסה חדשה	אבגי ראובן	ט'	1	1/2003
תכונות מנוע החיפוש Inter Text	אבגי ראובן	ט'	1	1/2003
החפשן של סנונית	אבולוף אוריאל	ה'	2	5/1999
טיפול בנושא השמות ביד ושם	אברהם אלכס	ט'	2	6/2003
Yad Vashem names and places index	אברהם אלכס	ט'	2	6/2003
חיפוש מידע ברשת	אדלשטיין קובי טל רפפורט	ח'	1	1/2002
מנוע חיפוש וניהול ידע בעברית	אופיר עידית זהבי יורם	ח'	2	6/2002
Full Txet - מנוע חיפוש בעברית	אורון שחר	ז'	1	1/2001
GUIdance : כלי ליצירת ממשק גרפי למערכת MF	אורנשטיין דרור	א'	2	10/1995
בחינת מודל ללמידה דרך הייפרטקסט	אחיטוב שמיר	ב'	1	5/1996
ארכיון צה"ל ומערכת הבטחון	אלגום אורי	ב'	1	5/1996
כלים לביצוע מחקר איטרטיבי	אמיתי יעל	ג'	1	2/1997
מנוע החיפוש RetrievalWare	אנדלמן נחמה צבי קמר	ט'	2	6/2003
ייצור אוטומטי של תיזאורי ומילונים דו לשוניים	ארד איריס	י'	2	6/2004
פיתוח מערכת טקסטואלית תומכת החלטה בסביבה מרובת פלטפורמות	אריאלי אהוד	ז'	1	1/2001
השוואת מערכות חיפוש של המאגר ה- Medline	בנחקון אלינור	ה'	2	5/1999
חבילת מוצרים לניתוח ואחזור שמות	בן אהרון דביר	ט'	2	6/2003
יציבות מידע ברשת	דר' בר-אילן יהודית	ו'	2	
ארכיון הכתבות הדיגיטליות בידיעות אחרונות	ברדוש יוסי	י'	1	1/2004
ממשק שפה טבעית בעברית למסכי נתונים יחסיים	גור אלי	א'	2	10/1995
היפרטקסט וקבלת החלטות	גוראון רן	ד'	1	2/1998
השוואה של מנועי חיפוש בעברית	גיל תומר	ח'	2	6/2002
פתרונות לאבטחת איכות תוכנה	גילעד זוהר	ד'	2	5/1998
טכנולוגיית ה- Push	גליקשטיין טליה	ג'	2	7/1997
הוצאה לאור אלקטרונית	דננברג אמיר	ג'	1	2/1997
הסבת ארכיון צה"ל ממחשב WANG למחשב HP תחת UNIX	דננברג בני	ד'	1	2/1998
חדשות מקבוצת הענין	דרורי עפר	א'	1	4/1995
		א'	2	10/1995
		ב'	1	5/1996
		ב'	2	11/1996
		ג'	1	2/1997
		ג'	2	7/1997
		ד'	1	3/1998
		ד'	2	5/1998
		ה'	1	1/1999
		ה'	2	5/1999
		ו'	1	1/2000
		ו'	2	6/2000
		ז'	1	1/2001
		ז'	2	6/2001
		ח'	1	1/2002
		ח'	2	6/2002
		ט'	1	1/2003
		ט'	2	6/2003

המשך אינדקס לפי שם מחבר (כרך א׳ עד כרך י׳ חוברת 2)

שם המאמר	שם המחבר	כרך	גליון מס.	תאריך
		י׳	1	1/2004
		י׳	2	6/2004
בניית מאגרי-מידע גדולים	דרורי עפר	ה׳	2	5/1999
ישום מערכת איחזור מידע טקסטואלי בשע״ם	דרורי עפר	א׳	1	4/1995
מנועי חיפוש באינטרנט	דרורי עפר	ג׳	1	2/1997
ניהול וארגון קבוצת ענין	דרורי עפר	ב׳	1	5/1996
עיצוב ממשק משתמש במערכות מידע	דרורי עפר	ה׳	1	1/1999
שילוב מערכות אחזור טקסט ומערכות מידע קונבנציונליות	דרורי עפר	ג׳	2	7/1997
הצגת תוצאת חיפוש בממשק משתמש במערכות אחזור טקסט - סקירת ספרות	דרורי עפר	ו׳	1	1/2000
הוספת מנוע אחזור לאתר אינטרנט	דרורי עפר	ו׳	2	6/2000
קריטריונים להשוואה בין מנועי חיפוש	דרורי עפר	ז׳	1	1/2001
שילוב בסיסי נתונים ומאגרי-מידע באתר ה- WEB בספריה ובמרכזי מידע	דרורי עפר	ז׳	2	6/2001
שילוב בסיסי נתונים ומאגרי-מידע באתר ה- WEB בספריה ובמרכזי מידע	דרורי עפר	ח׳	1	1/2002
מנועי חיפוש בעברית - רשימת ספקים	דרורי עפר	ח׳	1	6/2002
רשימת ספקים של מנועי אחזור בעברית (11.2002)	דרורי עפר	ט׳	1	1/2003
שימוש במילים נפוצות במסמך לאיתור נושא המסמך	דרורי עפר	ט׳	1	1/2003
ניהול תוכן - תכונות נדרשות	דרורי עפר	ט׳	1	1/2003
קריטריונים לבחירת מנוע אחזור טקסט - גרסה 2	דרורי עפר	ט׳	1	1/2003
מנוע חיפוש לשפה העברית	דרורי עפר	ט׳	2	6/2003
רשימת ספקים למנועי אחזור בעברית (3.2003)	דרורי עפר	ט	2	6/2003
מנועי אחזור טקסט בעברית - רשימת ספקים (גירסה 3.2004	דרורי עפר	י׳	2	6/2004
קריטריונים לבחירת מנוע אחזור טקסט - גירסה 3	דרורי עפר	י׳	2	6/2004
איתור נושא מסמך בצורה אוטומטית תוך שימוש במילים נפוצות	דרורי עפר	י׳	2	6/2004
Inter Text	הוך איציק	ד׳	1	2/1998
מערכת נוהלים בטכנולוגיית אינטרה-נט	הוך איציק	ד׳	2	5/1998
סיכום מצב קיים במערכות אחזור טקסט	דר׳ הנדזל רות	ב׳	2	11/1996
עיצוב ממשק משתמש ל- WEB	ווידנפלד צביקה	ד׳	2	5/1998
כיווני פיתוח באחזור טקסט TQL	וייזל גלעד	א׳	1	4/1995
ממשקים ויזואלים לתוצאות חיפוש	זמיר אורן עציוני אורן	ו׳	1	1/2000
ויזואליזציה של תוצאות חיפוש במערכות אחזור מידע	זמיר אורן	ז׳	2	6/2001
מגמות עתידיות בעולם איחזור המידע expetrext	ד״ר חנני אורי	א׳	1	4/1995
מנתונים לידע - MindCite	ד״ר חנני אורי	ט׳	2	6/2003
זיהוי תמידי של מידע ברשת האינטרנט	חן בועז	ו׳	1	1/2000
יין ואינטרנט	טיוטו מרק	ו׳	2	6/2000
מערכת לאיתור ישויות	יפת אביבה דרורי עפר	י׳	1	1/2004
תכונות מנוע החיפוש מורפיקס	ירדני לאורה	ט׳	1	1/2003
חיפוש טקסט מלא בעברית ובערבית - הבעיה והפתרון	ירדני לאורה	י׳	1	1/2004
תכונות מוצר לניהול הידע D2K.NET	כהן אייל	ט׳	1	1/2003
חיפוש תלוי הקשר, עקרונות ודוגמאות	כהן חנן	ח׳	1	1/2002
אחזור מסמכים משובשים	לבנה משה	ח׳	1	1/2002
המרת אתרים מעברית ויזואלית ללוגית	ליס אורלי	ט׳	2	6/2003
ארגון תוצאות חיפוש ממנועי אחזור באינטרנט	דר׳ לסט מרק	ח׳	2	6/2002
ממשק חלונאי אחד לטקסטים בסביבות עבודה שונות	מבשב ישראל טל כוכבה	ז׳	1	1/2001
תכונות מנוע החיפוש WizDoc	מידן אברהם	ט׳	1	1/2003

המשך אינדקס לפי שם מחבר (כרך א' עד כרך י' חוברת 2)

שם המאמר	שם המחבר	כרך	גליון מס.	תאריך
WizDoc - מנוע חיפוש לפי משמעויות בעברית ובאנגלית	מידן אברהם	ט'	2	6/2003
תכונות מנוע החיפוש Fast	מימוני אלון	ט'	2	6/2003
ארכיטקטורה של מנוע החיפוש Fast Search	מימוני אלון	ט'	2	6/2003
השוואה בין מימושים שונים של מורפולוגיה עברית ביישומי אחזור מידע טקסטואלי	מרגלית אפרים	י'	2	6/2004
מאגרי מידע משולבים טקסט ומידע חזותי	סגל בני	ב'	2	11/1996
המדריך למנועי חיפוש ברשת האינטרנט	סימסולו יניב	ה'	1	1/1999
עידון תהליכי חיפוש במאגרי מידע והצגתם למשתמש	סלייטר זיו חי-עזרא רחל	י'	2	6/2004
ניצול אופטימלי של מנועי חיפוש	פישל אריק	ח'	1	1/2002
מנוע חיפוש עברי במסדי נתונים מובנים	פלמון ערן	ה'	1	1/1999
אינטרנט	פריימן שלמה	ב'	1	5/1996
ההתפתחות המקבילה של מנועי חיפוש וספריות דיגיטליות	פרנק אריאל חנני אורי	ז'	2	6/2001
תכונות מנוע החיפוש Flair	פרנקל עפרה	ט'	2	6/2003
תזאורוס,הרעיון ושימושיו במערכות אחזור טקסט	צור מיכל	א'	2	10/1995
מנוע חיפוש כתשתית לאוטומציה של תהליכים ידניים במאגרים טקסטואליים	קולקו מיקי	ה'	1	1/1999
תכונות מנוע החיפוש XRS	קולקו מיקי	ט'	1	1/2003
אחזור טקסט בשמות באמצעות PowerMatcher	קולקו מיקי	י'	1	1/2004
אחזור מידע ומורפולוגיה של השפה הערבית	קמיר דרור	י'	2	6/2004
למה כל כך קשה לכתוב בעברית	רוזן יונתן	ה'	2	5/1999
לוח המפתחות העברי	רוזן יונתן	ה'	2	5/1999
ערכים מספריים לאותיות עבריות	רוזן יונתן	ה'	2	5/1999
TRS - מאחורי הקלעים - המרכיבים השונים של המערכת והיישומים האפשרים בה	רוזן פטר	א'	2	10/1995
טכנולוגיות ה- DTSearch	ריפתין אביב	ח'	2	6/2002
שימוש במטה-תגיות לשיפור וייעול הופעת אתרים במנועי חיפוש	רפפורט טל רשתי דודו	ח'	1	1/2002
ניהול ידע	רפפורט טל רשתי דודו	ח'	1	1/2002
עברית ברשת	רשתי דודו	ה'	2	5/1999
דגשים בנוגע לשילוב עברית במסמכי HTML	רשתי דודו ירחי איציק	ה'	2	5/1999
אחזור מידע ומנועי חיפוש	שוק אדם	ו'	1	1/2000
אחזור מידע המבוסס על תוכן תמונות במסמך	שטרן יוני	ג'	2	7/1997
ניהול תוכן במשרד במקר המדינה - ספריה דיגיטלית	שמחוני אלה	י'	1	1/2004
סינון מידע בטכניקות מתקדמות	שפירא ברכה	ב'	2	11/1996
כלי לחיפוש שיתופי באינטרנט - Antworld	שפירא ברכה	ז'	2	6/2001
שיטות להתאמה אישית (פרסונליזציה) של תוכן	שפירא ברכה	י'	1	1/2004
Information Retrieval Interaction	Ingwersen peter	ט'	2	6/2003
Information Retrieval	Rijsbergen C.J. Van	ט'	2	6/2003
XML - המסלול המהיר לכלכלה החדשה	שרוטר גרט	ז'	2	6/2001
כלים לחיפוש מתקדם ברשת האינטרנט	שרון יוחאי	ה'	1	1/1999
בדיקת תוכנה נתמכת מחשב	שריג עידא	ד'	2	5/1998

אינדקס לכרכים א׳ עד י׳ עפ״י כותרים

אינדקס לפי שם מאמר (כרך א׳ עד כרך י׳ חוברת 2)

שם המאמר	שם המחבר	כרך	גליון מס.	תאריך
אבני בניין לניהול ידע	מטה גרופ	י׳	1	1/2004
אחזור טקסט בשמות באמצעות PowerMatcher	מיקי קולקו	י׳	1	1/2004
אחזור מידע המבוסס על תוכן תמונות במסמך	יוני שטרן אבי עזרא	ג׳	2	7/1997
אחזור מידע ומנועי חיפוש	אדם שוק	ו׳	1	1/2000
אחזור מידע ומורפולוגיה של השפה הערבית	דרור קמיר	י׳	2	6/2004
אחזור מסמכים משובשים	משב לבנה	ח׳	1	1/2002
אינטרנט	שלמה פריימן	ב׳	1	5/1996
איתור נושא מסמך בצורה אוטומטית תוך שימוש במילים נפוצות	עפר דרורי	י׳	2	6/2004
ארגון תוצאות חיפוש ממנועי אחזור באינטרנט	דר׳ מרק לסט	ח׳	2	6/2002
ארכיון הכתבות הדיגיטליות בידיעות אחרונות	יוסי ברדוש	י׳	1	1/2004
ארכיון צה״ל ומערכת הבטחון	אורי אלגום	ב׳	1	5/1996
ארכיטקטורה של מנוע החיפוש Fast Search	אלון מימוני	ט׳	2	6/2003
בדיקת תוכנה נתמכת מחשב	עידא שריג	ד׳	2	5/1998
בחינת מודל ללמידה דרך הייפרטקסט	שמיר אחיטוב	ב׳	1	5/1996
בניית מאגרי-מידע גדולים	עפר דרורי	ה׳	2	5/1999
דגשים בנוגע לשילוב עברית במסמכי HTML	דודו רשתי איציק ירחי	ה׳	2	5/1999
ההתפתחות המקבילה של מנועי חיפוש וספריות דיגיטליות	פרנק אריאל אורי חנני	ז׳	2	6/2001
הוספת מנוע אחזור לאתר אינטרנט	עפר דרורי	ו׳	2	6/2000
הוצאה לאור אלקטרונית	אמיר דננברג	ג׳	1	2/1997
החפשן של סנונית	אוריאל אבולוף	ה׳	2	5/1999
היפרטקסט וקבלת החלטות	רן גוראון	ד׳	1	2/1998
המדריך למנועי חיפוש ברשת האינטרנט	יניב סימסולו	ה׳	1	1/1999
המרת אתרים מעברית ויזואלית ללוגית	אורלי ליס	ט׳	2	6/2003
הסבת ארכיון צה״ל ממחשב WANG למחשב HP תחת UNIX	בני דננברג	ד׳	1	2/1998
השוואה של מנועי חיפוש בעברית	תומר גיל	ח׳	2	6/2002
השוואה בין מימושים שונים של מורפולוגיה עברית ביישומי אחזור מידע טקסטואלי	אפרים מרגלית	י׳	2	6/2004
השוואת מערכות חיפוש של מאגר ה- Medline	אלינור בנחקון	ה׳	2	5/1999
הצגת תוצאת חיפוש בממשק משתמש במערכות אחזור טקסט - סקירת ספרות	עפר דרורי	ו׳	1	1/2000
ויזואליזציה של תוצאות חיפוש במערכות אחזור מידע	זמיר אורן	ז׳	2	6/2001
זיהוי תמידי של מידע ברשת האינטרנט	בועז חן	ו׳	1	1/2000
חבילת מוצרים לניתוח ואחזור שמות	דביר בן אהרון	ט׳	2	6/2003
חדשות מקבוצת הענין	עפר דרורי	א׳	1	4/1995
		א׳	2	10/1995
		ב׳	1	5/1996
		ב׳	2	11/1996
		ג׳	1	2/1997
		ג׳	2	7/1997
		ד׳	1	2/1998
		ד׳	2	5/1998
		ה׳	1	1/1999
		ה׳	2	5/1999
		ו׳	1	1/2000
		ו׳	2	6/2000
		ז׳	1	1/2001
		ז׳	2	6/2001

המשך אינדקס לפי שם מאמר (כרך א׳ עד כרך י׳ חוברת 2)

שם המאמר	שם המחבר	כרך	גליון מס.	תאריך
		ח׳	1	1/2002
		ח׳	2	6/2002
		ט׳	1	1/2003
		ט׳	2	6/2003
		י׳	1	1/2004
		י׳	2	6/2004
חיפוש טקסט מלא בעברית ובערבית - הבעיה והפתרון	ליאורה ירדני	י׳	1	1/2004
חיפוש מידע ברשת	אדלשטיין קובי וטל רפפורט	ח׳	1	1/2002
חיפוש תלוי הקשר	חנן כהן	ח׳	1	1/2002
טיפול בנושא השמות ביד ושם	אלכס אברהם	ט׳	2	6/2003
טכנולוגיות ה- DTSearch	אביב ריפתין	ח׳	2	6/2002
טכנולוגיית ה- Push	טליה גליקשטיין	ג׳	2	7/1997
יין ואינטרנט	מרק טויטו	ו׳	2	6/2000
ייצור אוטומטי של תיזאורי ומילונים דו לשוניים	איריס ארד	י׳	2	6/2004
יציבות מידע ברשת	דר׳ יהודית בר-אילן	ו׳	2	6/2000
ישום מערכת איחזור מידע טקסטואלי בשע״ם	עפר דרורי	א׳	1	4/1995
כיווני פיתוח באחזור טקסט TQL	גלעד וייזל	א׳	1	4/1995
כלי לחיפוש שיתופי באינטרנט - Antworld	שפירא ברכה	ז׳	2	6/2001
כלים לביצוע מחקר איטרטיבי	יעל אמיתי	ג׳	1	2/1996
כלים לחיפוש מתקדם ברשת האינטרנט	יוחאי שרון	ה׳	1	1/1999
לוח המפתחות העברי	יונתן רוזן	ה׳	2	5/1999
למה כל כך קשה לכתוב בעברית	יונתן רוזן	ה׳	2	5/1999
מאגרי מידע משולבים טקסט ומידע חזותי	בני סגל	ב׳	2	11/1996
מגמות עתידיות בעולם איחזור המידע expetrext	ד״ר אורי חנני	א׳	1	4/1995
ממשק חלונאי אחד לטקסטים בסביבות עבודה שונות	ישראל מבשב כוכבה טל	ז׳	1	1/2001
ממשק שפה טבעית בעברית למסכי נתונים יחסיים	אלי גור	א׳	2	10/1995
מנועי אחזור טקסט בעברית - רשימת ספקים (גירסה 3.2004)	עפר דרורי	י׳	2	6/2004
מנוע חיפוש וניהול ידע בעברית	עידית אופיר יורם זהבי	ח׳	2	6/2002
מנוע חיפוש כתשתית לאוטומציה של תהליכים ידניים במאגרים טקסטואליים	מיקי קולקו	ה׳	1	1/1999
מנוע חיפוש לשפה העברית	עפר דרורי	ט׳	2	6/2003
מנוע חיפוש עברי במסדי נתונים מובנים	ערן פלמון	ה׳	1	1/1999
מנוע האחזור Inter Text - גרסה חדשה	ראובן אבגי	ט׳	1	1/2003
מנוע החיפוש RetrievalWare	נחמה אנדלמן וצבי קמר	ט׳	2	6/2003
מנועי חיפוש באינטרנט	עפר דרורי	ג׳	1	2/1997
מנועי חיפוש בעברית - רשימת ספקים	עפר דרורי	ח׳	2	6/2002
מנתונים לידע - MindCite	אורי חנני	ט׳	2	6/2003
מערכת לאיתור ישויות	אביבה יפת עפר דרורי	י׳	1	1/2004
מערכת נוהלים בטכנולוגיית אינטרה-נט	איציק הוך	ד׳	2	5/1998
ממשקים ויזואלים לתוצאות חיפש	אורן זמיר אורן עציוני	ו׳	1	1/2000
ניהול וארגון קבוצת ענין	עפר דרורי	ב׳	1	5/1996
ניהול ידע	טל רפפורט דודו רשתי	ח׳	1	1/2002
ניהול תוכן - תכונות נדרשות	עפר דרורי	ט׳	1	1/2003
ניהול תוכן במשרד מבקר המדינה	אלה שמחוני	י׳	1	1/2004

המשך אינדקס לפי שם מאמר (כרך א' עד כרך י' חוברת 2)

שם המאמר	שם המחבר	כרך	גליון מס.	תאריך
ניצול אופטימלי של מנועי חיפוש	אריק פישל	ח'	1	1/2002
סיכום מצב קיים במערכות אחזור טקסט	רות הנדזל	ב'	2	11/1996
סינון מידע בטכניקות מתקדמות	ברכה שפירא	ב'	2	11/1996
עברית ברשת	דודו רשתי	ה'	2	5/1999
עידון תהליכי חיפוש במאגרי מידע והצגתם למשתמש	זיו סלייטר רחל חי-עזרא	י'	2	6/2004
עיצוב ממשק משתמש במערכות מידע	עפר דרורי	ה'	1	1/1999
עיצוב ממשק משתמש ל- WEB	צביקה ווידנפלד	ד'	2	5/1998
ערכים מספריים לאותיות עבריות	יונתן רוזן	ה'	2	5/1999
פיתוח מערכת טקסטואלית תומכת החלטה בסביבה מרובת פלטפורמות	אריאלי אהוד	ז'	1	1/2001
פתרונות לאבטחת איכות תוכנה	זוהר גילעד	ד'	2	5/1998
קיבוץ שאילתות במנועי חיפוש	גדי גולדרינג ואיתן פאר	י'	1	1/2004
קריטריונים לבחירת מנוע אחזור טקסט - גרסה 2	עפר דרורי	ט'	1	1/2003
קריטריונים לבחירת מנוע אחזור טקסט - גרסה 3	עפר דרורי	י'	2	6/2004
קריטריונים להשוואה בין מנועי חיפוש	עפר דרורי	ז'	1	1/2001
רשימת ספקים למנועי אחזור בעברית (3.2003)	עפר דרורי	ט'	2	6/2003
רשימת ספקים של מנועי אחזור בעברית (11.2002)	עפר דרורי	ט'	1	1/2003
שבעה צעדים לניהול ידע	דוד יוקלסון	י'	1	1/2004
שיטות להתאמה אישית (פרסונליזציה) של תוכן	ברכה שפירא	י'	1	1/2004
שילוב בסיסי נתונים ומאגרי מידע באתר ה- Web בספריה ובמרכזי מידע	עפר דרורי	ז'	2	6/2001
שילוב בסיסי נתונים ומאגרי - מידע ב- Web	עפר דרורי	ח'	1	1/2002
שילוב מערכות אחזור טקסט ומערכות מידע קונבנציונליות	עפר דרורי	ג'	2	7/1997
שימוש במטה-תגיות לשיפור הופעת אתרים במנועי חיפוש	רפפורט טל דודו רשתי שולה גורן	ח'	1	1/2002
שימוש במילים נפוצות במסמך לאיתור נושא המסמך	עפר דרורי	ט'	1	1/2003
תזאורוס, הרעיון ושימושיו במערכות אחזור טקסט	מיכל צור	א'	2	10/1995
תכונות מוצר לניהול הידע D2K.NET	אייל כהן	ט'	1	1/2003
תכונות מנוע החיפוש מורפיקס	ליאורה ירדני	ט'	1	1/2003
תכונות מנוע החיפוש Wiz.Doc	אברהם מידן	ט'	1	1/2003
תכונות מנוע החיפוש XRS	מיקי קולקו	ט'	1	1/2003
תכונות מנוע החיפוש Fast	אלון מימוני	ט'	2	6/2003
תכונות מנוע החיפוש Flair	עפרה פרנקל	ט/	2	6/2003
Information Retrieval	Rijsbergen C.J. Van	ט'	2	6/2003
Information Retrieval Interaction	Peter Ingwersen	ט'	2	6/2003
תכונות מנוע החיפוש Inter Text	ראובן אבגי	ט'	1	1/2003
Full Text - מנוע חיפוש בעברית	שחר אורון	ז'	1	1/2001
GUIdance : כלי ליצירת ממשק גרפי למערכת MF	דרור אורנשטיין	א'	2	10/1995
Inter Text	איציק הוך	ד'	1	2/1998
TRS - מאחורי הקלעים - המרכיבים השונים של המערכת והיישומים האפשרים בה	פטר רוזן	א'	2	10/1995
WizDoc - מנוע חיפוש לפי משמעויות בעברית ובאנגלית	אברהם מידן	ט'	2	6/2003
XML - המסלול המהיר לכלכלה החדשה	שרוטר גרט	ז'	2	6/2001
XML והשלכותיו על בסיסי נתונים ואחזור טקסט	אבגי ראובן	ו'	2	6/2000
Yad Vashem names and places index	Alex Avraham	ט'	2	6/2003