חיפוש עברי: לראשונה בקוד פתוח. אתגרים, פתרונות והתמודדויות אחרות.

איתמר סין-הרשקו

מבוא

בעולם עתיר מידע, אחזור טקסט יעיל הוא מהבעיות הטכנולוגיות המורכבות ביותר. חברות וארגונים נעזרים בכלים טכנולוגיים שונים כדי לאפשר חיפוש במאגרים שברשותם, אך להבדיל מקטלוג ידני של החומר, אין בכך ערובה לאיכות התוצאות. שימוש באינדקס (inverted index) – מלבד מהירות אחזור גבוהה – מאפשר גם אומדן רלוונטיות ודירוג התוצאות לפי פרמטרים מותאמים אישית.

איכותו של מנוע לאחזור טקסט נמדדת לפי כמות התוצאות ה"טובות" ברשימת התוצאות, מיקומן ביחס לתוצאות הלא רלוונטיות, ולפי מספר התוצאות הרלוונטיות שלא אוחזרו כלל. בעולם מושלם, מנוע חיפוש יציג עבור כל שאילתא את כל המסמכים הרלוונטיים, ורק אותם, בדירוג רלוונטיות נכון.

מאז פריצתו של האינטרנט לחיינו, המחקר בתחום הואץ והגיע להישגים מרשימים. מנועים מורכבים פותחו (חלקם כקוד פתוח), ועבודתם עם שפות רבות נבחנה. עבור השפה העברית מעולם לא נעשתה עבודה בסדר גודל מקביל, אף שבשנים האחרונות ניכרת מודעות גוברת לעניין. בעברית האתגר גדול הרבה יותר מבשפות אחרות, עקב מורפולוגיה עשירה שגורמת לעמימות רבה.

מטרתנו בפרוייקט HebMorph היא לספק כלים לביצוע אומדני רלוונטיות בעברית, ולבדוק באמצעותם גישות שונות לחיפוש עברי שיפותחו גם הן במסגרת הפרויקט. כיוון שהעבודה כולה היא בקוד פתוח, תהיה לכל מפתח – לראשונה – גישה לכלי חיפוש איכותיים עבור טקסטים בעברית.

במסמך זה ניתן סקירה מקיפה על הנושא: הצגת הבעיה, דרכי פתרון אפשריים, ומה אנו רואים כפתרון המלא והמקיף ביותר לבעיה.

הצגת הבעיה

כיצד אינדקס עובד?

הדרך הטובה ביותר לאפשר חיפוש טקסטואלי מהיר, היא ע"י שימוש באינדקס מהופך (inverted index). תהליך יצירת האינדקס הינו פשוט למדי: איסוף כל המילים שבקורפוס, מיונם בסדר אלפביתי, כאשר כל מילה מכילה מידע לגבי כל המקומות שבהם היא מופיעה (מספר סידורי של מסמך, למשל).

כך מתקבלת לה דרך יעילה מאד לחיפוש טקסטים במאגרים גדולים: במקום מעבר על אלפי או מיליוני מסמכי טקסט מפוזרים, מתבצע חיפוש מילוני מהיר בקובץ אינדקס אחד.

עבודה עם אינדקס בשפות שונות

כיוון שהחיפוש הוא מילוני ומדויק, מקובל לבצע נורמליזציה של הטקסט כדי לוודא שגם הטיות שונות של מילה ייחשבו כמופע רלוונטי, בהתאם לתכונות השפה. כך למשל באנגלית, שימוש בכלי stemming (מציאת גזע המילה) כמו Porter יאפשרו לחיפוש עבור המילה look לאחזור מופעים של looked, looking, looker, או לבטל אבחנה בין רבים ליחיד (books → book), וכן הלאה.

קיימות גם שיטות נורמליזציה אחרות באנגלית, כמו s-stemmer (המסתפקת בהשמטת הסיומת האנגלית המסמלת רבים בלבד), soundex (השוואת צורות הגייה) ועוד.

אחזור מידע בעברית

במרבית השפות בעולם איות המילים לא משאיר הרבה מקום לספק באשר למשמעותה האמיתית של מילה נתונה, והוא הדין גם לעברית מנוקדת. כאשר כל מילה היא ייחודית, ורק לעתים נדירות קיימת עמימות בנוגע למשמעותה האמיתית, כל שנותר הוא לכותבה לאינדקס כמות שהיא (או לאחר נורמליזציה).

הכתיב העברי חסר הניקוד בעצם הגדרתו משמיט תנועות מהכתיבה, ולכן מונע את זיהוי המשמעות האמיתית של מילים רבות בשפה כשהן מובאות בלא הקשר. כיוון שלכ-50% מכלל המילים קיימות שתי משמעויות ויותר, נוצרת אי-ודאות לגבי המונח (term) הנכון שעלינו להכניס לאינדקס. לדוגמא, האם הכוונה במילה "שני" היא לשם של ילדה, לצורה של המספר 2 (second, או סמיכות), לציווי האומר "לכי לישון", או לְ-שוני (difference) בכתיב חסר.

עמימות זו מקשה גם על הפרדת חלקי מילה. אותיות השימוש מש"ה וכל"ב משמשות כתחיליות עבור אחוז ניכר מהמילים בעברית, ואינן נחשבות כחלק מהמילה (לדוגמא: בית ← בבית, לבית, שבבית). כיוון שהאינדקס ממוין אלפביתית, מונחים כאלו לא יימצאו כלל ללא השמטת התחיליות. קיימים מקרים רבים בהם לא ניתן לדעת ללא הקשר וללא ניקוד האם האות משמשת כתחילית, או שהיא למעשה חלק אינטגרלי מהמילה. לדוגמא: "רותי פספסה את הרכבת" לעומת "הרכבת המוצר מסובכת להפליא", ולעתים יהיו מקרים עמומים אף יותר (למשל: שבתו).

עמימות יכולה להשפיע גם על הפרדת סופיות. המילה "חבלה" לדוגמא, יכולה להיות עם מפיק ה"א (החבל שלה), או להוות פועל (ביצעה חבלה, בפיעל או בפעל).

בעוד לכתיב המנוקד חוקיות ברורה שמונעת משתי מילים שונות בתכלית להיכתב באותה צורה, החוקים לכתיב חסר הניקוד נתונים לעתים קרובות במחלוקת [1]. גם במקומות בהם ישנה הכרעה ברורה, לא תמיד האיות התקני בשימוש נרחב, ולעתים הצורה השגויה היא הפופולארית יותר (למשל, הכתיב התקין של המילה "אימא" הוא עם יו"ד...). על כן, טקסטים מאייתים את אותה המילה בכמה צורות שונות; לעתים קרובות הדבר קורה אף בגוף אותו מסמך שנתחבר ע"י אותו המחבר. כך מתווספת לה שכבת עמימות נוספת, והמילה "שירות" (service) שעשויה להיכתב גם כ-"שרות" (they sing), עלולה להיות מוחלפת עם "שיירות" (convoys); או, לחלופין, אחשורוש ואחשוורוש ייחשבו לשני אנשים שונים.

חלק מכללי האיות האמורים דנים על צורת הכתיב הנכונה למילים שאולות. מבחינת הציבור שאלות כמו האם כותבים שבדיה או שוודיה, טורקיה או תורכיה, פריס, פריז או פאריז – מעולם לא קיבלו תשובה, ולכן הם עשויים לבחור כל כתיב בו יחפצו.

יצירת רשימה מלאה של "מילות עצירה" גם היא משימה בלתי אפשרית עקב עמימות. מילים כמו "אשר", "כדי" ו"אף" – שבצורתן הפשוטה נראות כמילות עצירה לכל דבר, יכולות גם להיות מילים בעלות משמעות, לעתים אף חשובה (למשל, אם שם נשוא מאמר הוא אשר). הבעיה ההפוכה קיימת עבור צירופי מילים כגון "על ידי", "אי פעם", "אף על פי", "שום דבר" – שם מילים שעשויות להיות רבות משמעות בכל הקשר אחר, נטולות כל משמעות בתצורתן זו. בדומה לכך, גם בסמיכויות כדוגמת "פי התהום" מילים עשויות לאבד את משמעותן המקורית.

נזכיר גם את ראשי-התיבות העבריים, שלמעשה הופכות את תו הגרשיים כאות מן המניין, לפחות מבחינת המערכות הממוחשבות הנידונות (קריטי לתהליכי חיתוך טקסט - tokenization). הדבר נכון גם לתו הגרש, שמסמן גם הוא קיצורי מילים ומשנה את הגייתן של האותיות חצ"ץ ג"ז. במקרים מסוימים גם הגרש עלול ליצור עמימות – למשל המילה "אינצ' ", שלא ברור אם כוונתה למידה האנגלית או לקיצור של "אינציקלופדיה".

דיאלקטים עבריים שונים עשויים להציג מילים שאינן מזוהות (OOV), או לתת משמעות אחרת, בלתי-צפויה למילה ידועה ("חמר" כמשל; בימינו הכוונה לסוג אדמה, בעוד בטקסטים תלמודיים ובמשנה הכוונה היא ליין).

רובם המוחלט של הטקסטים היום אינם מנוקדים, ועבור מרבית המקרים שהובאו גם אדם דובר עברית רהוטה לא יוכל להכריע את העמימות ללא הקשר.

דרכי פתרון

מה לאנדקס?

זו השאלה החשובה מכולן. ההחלטה על זהותה של "יחידת האינדקס", המחרוזת שתישמר בקובץ האינדקס ותשמש לחיפושים, היא אבן היסוד למנוע חיפוש טוב, יעיל ומהיר. עבור השפה העברית, קיימות האפשרויות הבאות:

1. **המילה המקורית**. בהתחשב בבעיות שהועלו בסעיף הקודם, ניתן לשלול את האפשרות הזו בקלות. שימוש בתווים חופשיים (wildcards) עלול אף להחמיר את המצב, שכן חיפוש עבור *שיר* יאחזר "השירים", "השירות" ו-"שיריים".
2. **השורש העברי של המילה**. שיטה זו קרוב לוודאי תזיק יותר מאשר תועיל, שהרי כל שורש בן 3 אותיות יכול לשמש בצורות שונות רבות, המנותקות כל הקשר זו מזו.
3. **לֶמֶה (lemma)**. תבנית מילה בהטייתה הבסיסית (למשל דלתותינו ← דלת).
4. **פסבדו-למה, או גזע מילה (stem)**. חיתוך מלאכותי של מורפולוגיה רציפה מהמילה (התחיליות מש"ה וכל"ב והסיומות -ים, -ות ושכדוגמתן), באמצעות מערכת כללים שיכולה להיות פשוטה או מורכבת. התוצר של פעולה זו לא בהכרח יהיה בעל משמעות אמיתית או יפיג את העמימות שהייתה במקור.

עיבוד שפה טבעית (NLP) עבור עברית

ההנחה הרווחת היא שכדי לשמור על רלוונטיות של תוצאות חיפוש, עלינו למצוא את הלמה הנכונה עבור המילה הנתונה ואותה יש לשמור לאינדקס. מציאת למה כרוכה בניתוח מורפולוגי של המילה והפגת העמימות במידה וקיימת. תהליך הפגת העמימות המלא כרוך במספר פעולות, כאשר הראשונה שבהן היא לאחזר את כל הלמות האפשריות, וזאת ניתן לעשות באחת משתי דרכים:

1. **באמצעות מילון**, שנוצר באופן ידני, מתוך קורפוס או מתוך רשימת מילים בסיסית שהורחבה באמצעות אלגוריתמים. החיפוש במילון עשוי להתבצע מספר פעמים, כדי לקחת בחשבון תחיליות או הבדלי איות.
2. **אלגוריתמים מורפולוגיים**, שמזהים באמצעות מערכת חוקים למות אפשריות (לרוב על ידי זיהוי גזירה ממשקלים ובניינים). כיוון שהשיטה אלגוריתמית, ניתן להתחשב באמצעותה בהבדלי איות שונים.

ההבדל בין שתי הגישות תלוי באיכות האלגוריתם או המילון. הקריטריונים להשוואה:

- דיוק מורפולוגי, כלומר מה אחוז המילים בשפה שמזוהה בצורה נכונה, וכיצד תופעות מורפולוגיות מיוחדות מטופלות (למשל: broken plurals, שורשים בני 4-5 עיצורים, סיכול עיצורים וכד').

- הטיפול במילים שאולות, שמות אנשים ומקומות, או סלנג. זהו החלק בשפה שלעולם אינו חדל מלגדול.
- יכולת להתעלם (או: "לסבול") מצורות איות שונות.
- הפגת עמימות מובנית (אחוז הטעויות, הפגת עמימות קונטקסטואלית / POS, יכולות דירוג).

ברוב המקרים, פתרונות מבוססי מילון יספקו למות אמינות יותר, אך יש לבדוק את היקף כיסוי השפה שלהם באמצעות קורפוס. כמו"כ, מילונים יש לדאוג לעדכן באופן שוטף עם מילים חדשות, מילים שאולות, סלנג ושמות אנשים ומקומות. מילונים רבים יאפשרו לקבל מידע מורפולוגי מועיל, באמצעותו ניתן לבצע הרחבות של החיפוש (למשל, מילים נרדפות או הטיות מקבילות).

בעוד חיפוש מילוני מתבצע בצורה מדויקת לפי מפתח נתון, קל יותר להתאים אלגוריתם להתמודד עם תופעות מורפולוגיות שונות, כולל צורות איות שונות. אך מערכת חוקים כזו גם עמידה פחות בפני מקרים מורפולוגיים שאינם מוכרים לה, ועשויה לספק למות שגויות עבורן. מילים שאולות וסלנג, למשל, לא יזוהו כלל (במקרה הטוב) בהיעדר חוק מתאים עבורם.

מילונים לרוב תלויים בקבצים חיצוניים, או במסד נתונים, בעוד אלגוריתם מורפולוגי מבוסס על מערכת חוקים שניתן ליישמה בקוד בלבד, ללא כל תלות חיצונית.

מילונים ניתן ליצור באופן ידני (עם עזרה מכלים אוטומטיים), או באמצעות סריקת המילים מקורפוס גדול. מילונים מהסוג הראשון לרוב מכילים מידע מורפולוגי על כל מילה, כולל מידע על התחיליות שחוקיות עבורה. מילונים שנוצרו מגירוד של מאגר טקסט לרוב מכילים מידע סטטיסטי שעשוי לסייע בתהליך הפגת העמימות. איכותו של תהליך סטטיסטי כזה תלוי בכמות הטקסט שנסרק, בהקשר שלו ובעקביות האיות; עבור קורפוס טוב, היא עשויה לתאום לזו של אלגוריתם מורפולוגי.

בכל שיטה שנבחר, בשלב זה עשויות להיות בפנינו יותר מלמה אחת. ניתן לנסות ולסנן אותן לפי קריטריונים יבשים, למשל דירוג לסבירות הלמה, במידה וניתן ע"י השיטה בה בחרנו. לשלילת למות והפגת עמימות נוספת ניתן להשתמש בדרכים נוספות, רובן מסתמכות על הקשר המילה המקורי [2].

ניתוח משפט מלא (POS), בין אם בכלים סטטיסטיים או תחביריים, הוא הדרך היחידה להפגת עמימות מדויקת עבור טקסטים בעברית. משימה זו אינה פשוטה כלל (עבודה בתחום נעשית כבר מספר שנים), וגם אז לא לכל מילה תהיה משמעות מוחלטת אחת. הנה דוגמה: "המראה של מטוסים ריקים [...]".

אחזור טקסט מבוסס NLP

לאחר הפגת העמימות – מלאה או חלקית – מערכות לאחזור טקסט עשויות לבצע מספר פעולות על הלמות שהתקבלו לפני שישמרו אותן באינדקס:

1. כאשר ישנן מספר למות שונות, סינונן לפי דירוג או מידע סטטיסטי. במידה וגם לאחר סינון קיימת יותר מלמה אפשרית אחת, כל הלמות יישמרו באותו מיקום באינדקס.
2. טיפול במילים ללא למה (OOV). הטיפול יכול לכלול השוואה לרשימות מילים ידועות (כתובות או שמות פרטיים, למשל) או הסרת תחיליות וסופיות לפי חוקיות שרירותית. מערכות מסוימות עשויות לשמור את המונח כמו שהוא ללא כל טיפול נוסף.
3. הסרת "מילות עצירה" ו"מילות רעש". ניתוח קונטקסטואלי עשוי לשמש להפגת עמימות.
4. הרחבת מונחים, כמו soundex ומילים נרדפות, באמצעות כלים חיצוניים.

שיטות נוספות לאחזור טקסט

מחקרים רבים הוכיחו כי ניתוח מורפולוגי מלא אינו תורם משמעותית להשגת תוצאות חיפוש רלוונטיות, ולעתים אף פוגע בכך. שיטות כמו light-stemming, קיצוץ מילים (word truncation), n-grams, skipgrams ותת-סוגים שלהן הוכחו כיעילות מאד, ובמקרים מסוימים אף יעילות יותר מניתוח מורפולוגי [3]. גם במקרים עבורם שיטות אלו אינן יותר יעילות, לעתים קרובות הן עדיין יועדפו על פני כלים מורפולוגיים כבדים. חסרונותיהם העיקריים הם גודל האינדקס, ועבור חלקן זמן חיפוש ארוך יותר.

הסיבה לכך היא, ככל הנראה, שרלוונטיות באחזור מושפעת מנטייה סטטיסטית וכפילויות מונחים, וכדי להשיג זאת אין צורך אמיתי בהבנת השפה. כמו כן, רלוונטיות באחזור יכולה להיפגע גם מהפגת עמימות שגויה, וכזו ברור שקיימת בכל כלי מורפולוגי מתקדם ככל שיהיה.

בעוד הוכח כי מורפולוגיה אינה נדרשת כדי להשיג תוצאות אחזור טובות בשפות לועזיות, המורפולוגיה הייחודית והמורכבת של השפות השמיות לכאורה הופכת אותן ליוצאות מן הכלל. מחקרים דומים שנעשו על השפה הערבית הראו גם הם כי 4-grams ו-light-stemming מתפקדות טוב מאד, אף יותר ממנתח צורני [4][5]. כיוון שהמורפולוגיה הערבית דומה בהרבה מובנים לעברית, ניתן בהחלט לצפות לתוצאות דומות בעברית.

כיוון שתהליך מציאת הלמה הנכונה נסמך על הקשר המילים בטקסט המקור, ייתכן שגם מערכות אחזור מורפולוגיות עם ניתוח קונטקסטואלי מוצלח נתקלות במכשול בעת פענוח שאילתות בטקסט חופשי. תהליך הלמטיזציה עבור שאילתות עלול לייצר למות שגויות עקב העדר קונטקסט מועיל, מה שיגרום לשאילתה להיות שגויה ולפגוע ברלוונטיות התוצאות. גישות לא מורפולוגיות לא ייתקלו בבעיה כזו.

שיטת האחזור האופטימלית עבור עברית

כיוון שבשורה התחתונה עברית וערבית אינן זהות מורפולוגית, שיטות אחזור שעובדות טוב בערבית אינן מוכרחות להיות יעילות גם עבור טקסטים בעברית. עם זאת, ללא סביבת בדיקה שתאפשר לפתח ולנסות גישות שונות, לעולם לא נדע בוודאות.

פרוייקט הקוד הפתוח HebMorph נועד למטרה זו בדיוק, ולמיטב ידיעתנו לא קיימת כיום אף יוזמה דומה.

הפתרון

HebMorph

רוב הפתרונות הקיימים כיום לחיפוש עברי מסחריים, וכפי שהראינו ישנו גם ספק סביר לגבי עדיפותם על פני שיטות אחרות, זולות יותר. המטרה בעבודה על HebMorph כפולה: לקדם את תחום החיפוש העברי ע"י חיפוש תמידי אחר שיטת חיפוש יעילה יותר, והנגשת הכלים והידע לכל המעוניין במסגרת רישיון קוד פתוח.

במהלך העבודה על הפרויקט, יפותחו גישות שונות לטיפול בטקסטים עבריים עבור מערכות לאחזור מידע, ואלו ייבדקו באמצעות כלי בדיקה ייעודיים במסגרת פרויקט OpenRelevance של ארגון התכנה החופשית Apache [6]. מטרתו הסופית של הפרויקט היא להצביע על שיטות האחזור הטובות ביותר עבור עברית, ולספק אותן לכל המעוניין בקוד פתוח, עם אינטגרציה פשוטה לכל טכנולוגיית IR ועל כל פלטפורמה.

בזכות גמישותה ועוצמתה, נבחרה Apache Lucene Java [7] להיות המערכת העיקרית איתה נעבוד בשלב זה. נכון לשעת כתיבת שורות אלו מכיל הפרויקט גם כלי למטיזציה (lemmatizer) המבוסס על hspell [8], והוא כיום הכלי המרכזי בספריה המשמש לחיפוש עברי.

סטטוס נוכחי ותוכניות להמשך

שימוש ב-HebMorph כחלופה לברירות מחדל של מערכות IR מומלץ גם בתצורתו הנוכחית של הפרוייקט. ניתן להתרשם מהספריה דרך כלי המדגים את יכולותיה, ובאמצעות Lucene.Net מאפשר חיפוש עברי מורפולוגי בויקיפדיה העברית [9]. החיפוש המורפולוגי מסוגל לאפשר גם חיפוש מדויק, ותהליך סינון הלמות גם הוא ניתן להתאמה אישית [10].

כדי לקבל מידע נוסף, שמטבע הדברים יהיה טכני יותר ולכן אינו מובא כאן, ניתן ליצור קשר דרך רשימת התפוצה של הפרויקט (ראה בהמשך).

העבודה ממשיכה להתקדם במספר מישורים שונים:

1. שיפור כלי הטוקניזציה (חיתוך מחרוזת למילים)
2. שיפור הכלי המורפולוגי, במספר דרכים:
    a. הרחבת המילון (תיקון שגיאות, הוספת מילים חסרות, תמיכה בסבירויות).
    b. שיפורים במנגנון המאפשר "סבילת" איותים שונים.
    c. טיפול מורכב יותר במילים לא מזוהות.
    d. סינון חכם יותר של "מילות עצירה".
3. יצירת כלים שיאפשרו דירוג יעילות גישות אינדוקס שונות (דרך OpenRelevance).
4. פיתוח גישות נוספות, והשוואתן לגישות אחרות.
5. זמינות משפות אחרות (Java ועוד), ומכלי IR נוספים (Xapian, CLucene ועוד).

הצטרפו אלינו!

כדי שנוכל להשתפר, אנו צריכים פידבק. כל מי שהנושא מעניין אותו או שביכולתו לעזור, מוזמן להצטרף – גם אם זהו איננו התחום בו הוא מתמחה.

רשימת התפוצה שלנו:

https://lists.sourceforge.net/lists/listinfo/hebmorph-thinktank

הקוד זמין לכל דורש תחת רשיון GPLv2:

http://github.com/synhershko/HebMorph

עדכונים שוטפים על התקדמות הפרוייקט:

http://www.code972.com/blog/hebmorph/ / http://www.code972.com/blog/tag/hebmorph/

הפניות

[1] ראה למשל נדב הראל, "סוגיות בכתיב חסר הניקוד": http://hspell.ivrix.org.il/niqqudless.pdf.

[2] ראה:

"Previous work" in "A Corpus Based Morphological Analyzer for Unvocalized Modern Hebrew", by Alon Itai and Erel Segal, Department of Computer Science, Technion - Israel Institute of Technology, Haifa, Israel. http://www.mt-archive.info/MTS-2003-Itai.pdf.

[3] http://www.eecs.qmul.ac.uk/~christof/html/publications/inrt142.pdf

[4] http://web.jhu.edu/bin/q/b/p75-mcnamee.pdf

[5] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.87.877&rep=rep1&type=pdf

[6] http://lucene.apache.org/openrelevance/

[7] http://lucene.apache.org/java/

[8] http://hspell.ivrix.org.il/

[9] http://github.com/synhershko/BzReader

[10] http://www.code972.com/blog/2010/07/more-flexible-hebrew-indexing-hebmorph/