חיפוש בטקסט חופשי בעברית באמצעות שאילתות הכוללות כמה מילים

תוכנת WizDoc

ד"ר אברהם מידן

מה הדרך היעילה לחפש קטעים רלוונטיים (לדוגמה, תשובות לשאלות) בטקסט חופשי בעברית?

הדרך המקובלת ו"הפשוטה" ביותר היא: רושמים מילה, והתוכנה מחפשת את המחרוזת.

השיטה עובדת לא רע באנגלית, אבל בעברית היא יוצרת הרבה החטאות ואזעקות שווא, עקב הגורמים הבאים:

ניקוד: לדוגמה, "שבת" במובן "יום שבת", לעומת "שבת" במובן "השתתף בשביתה".

הטיות לשוניות: לדוגמה הם מכים, הוא הכה (רק האות "כ" משותפת)

אותיות שבשפות אחרות הן מילות יחס עצמאיות מהוות חלק מהמחרוזת: לדוגמה, "מועד" במובן חג, לעומת "מועד העובדים".

אין כתיב אחיד, לדוגמה שיפור לעומת שפור.

אפשרות אחת להתגבר על הבעיות שלעיל היא: בשלב יצירת האינדקס התוכנה תמצא את המשמעות הנכונה של המחרוזת על סמך מילים סמוכות, לדוגמה "יום שבת" לעומת "העובד שבת", ובשלב החיפוש, לגבי כל מילה בשאילתה התוכנה תציג משמעויות אפשריות, והמשתמשים יבחרו את המשמעות הנכונה.

יישמנו את השיטה הזו בגרסה ישנה. החסרונות העיקריים של השיטה: (1) המשתמשים לא "אהבו" את הדרישה לבחור משמעות (במקרים רבים התקשו להבין על סמך מה מוצגות האפשרויות לבחירה). (2) כדי ללמד את התוכנה למצוא את המשמעות הנכונה (בשלב יצירת האינדקס) יש לאמן אותה על כמו גדולה מאוד של טקסט ולהשקיע סכום גבוה מאוד בשעות עבודה רבות של אנשים, שיציינו בכל מקום מה המשמעות הנכונה. לאור השוק המצומצם בישראל, הגענו למסקנה שההשקעה לא כדאית.

עברנו לשיטת החיפוש הבאה: מעודדים את המשתמשים לרשום בשאילת החיפוש כמה מילים, ותוך כדי רישום השאילתה, התוכנה מציגה תוצאות ב- autocomplete, כך שאם התוצאות הרצויות לא נמצאות, משנים קצת את השאילתה עד שמוצאים את התוצאה הרצויה.

רשימת התוצאות כוללת את כל הטקסטים (100 מילים) שעונים על הדרישות הבאות:

מתעלמים ממילות היחס ולגבי כל רצף של 1 עד 4 מלים (בשאילתת החיפוש ובטקסט), ומציגים את הטקסטים שעונים על אחד התנאים הבאים:

- כל המילים זהות ולפי הסדר
- כל המילים זהות למעט מקרה אחד של שיכול מילים

- כל המילים זהות למעט מילה אחת שחסרה, נוספה, או הוחלפה במילה אחרת.

כמו כן: מותר שמילה אחת בטקסט או בשאילתת החיפוש דומה פונטית (אך לא זהה) למילה המקבילה (בטקסט או בשאילתת החיפוש). זוג המילים נחשב כאילו היו זהות.

מילה נחשבת לדומה פונטית, כאשר:

מתעלמים מכל האותיות שהן אמות קריאה ("ו", "י", למעט אם הן אחת אחר השנייה או כפולות, "ה" ו- "א" בסוף מילה)

וגם מתקיים אחד התנאים הבאים:

- כל העיצורים זהים ולפי הסדר
- כל העיצורים זהים למעט מקרה אחד של חילוף אותיות
- כל העיצורים זהים למעט עיצור אחד שחסר, נוסף, או הוחלף.

קל לראות שיש דמיון בין שני החיפושים: מילות היחס מקבילות לאמות קריאה והמלים מקבילות לעיצורים. כלומר, בחיפוש לפי מלים מתייחסים לכל מלה (שאינה מלת יחס) כמו שבחיפוש לפי דמיון פונטי מתייחסים לעיצורים.

החיפוש לפי דמיון פונטי מתגבר על טעויות כתיב. שיטה אלטרנטיבית להתגבר על טעויות כתיב היא באמצעות האלגוריתם הבא: בשלב יצירת האינדקס מתייקים כל רצף של 3 או 4 אותיות (באמצעות חלון נע), ובשלב החיפוש, שוב שולפים את כל הרצפים של 3 או 4 אותיות, ומוצאים את המילים שכוללות רצפים זהים. השיטה הזו יעילה כאשר המילים ארוכות. בעברית השיטה אינה יעילה, כיוון שהמילים קצרות ויש הרבה טעויות באותיות כמו "ו" ו- "י" באמצע מילה.

התוצאות ב- autocomplete ממוינות לפי מידת הדמיון לשאילת החיפוש, לפי הסדר הבא:

- תוצאות שכוללות את כל רצף המילים בשאילתה
- תוצאות הכוללות את רצף המלים בשאילתה עם רווח של מילה אחת
- תוצאות הכוללות את כל רצף המלים, כאשר מילה אחת מתאימה בעקבות דמיון פונטי

האתגר: איך לבצע את החיפוש במהירות גם כאשר מסד הנתונים הוא ענק? (כי ללא חיפוש מהיר אין טעם ב- autocomplete).

שליפת מילים לפי דמיון פונטי מבוצעת לפי פטנט שלנו U.S. Patent No. 10,409,861. הרעיון המרכזי באלגוריתם: שומרים את כל הרצפים של העיצורים, כולל רצפים לאחר שיכול אותיות אחד, וכולל רצפים שבהם חסר עיצור אחד. מבצעים אותה פעולה לגבי הערך בשאילתה, ומחפשים ערכים שווים.

החיפוש בטקסט החופשי מבוצע באמצעות האלגוריתם suffix index. ניתן לקרוא ב- Wikipedia: https://en.wikipedia.org/wiki/Suffix_array

לאחר ההרצאה יוצגו חיפושים באמצעות WizDoc בקובץ העזרה של חשבשבת:

דוגמה לשאילתות

ניכוי

איך מנכים מס

איך מנכים מס במקור

אופטימיזציה של רכש

איך לחשב רווח ללקוח