

פרק 7

GAR - הזרקת ידע ארגוני לבינה המלאכותית

מטרות הלמידה

בסיום פרק זה תהיה מסולgilim:

- להבין לעומק את מנגנון GAR (noitareneG detnemguA-laveirteR) ואת היתרונות שהוא מביא לארגונים
- לתכנן ולהטמי מערכות GAR המותאמת לצרכי הארגון שלכם
- לשפר את דיקוק התשובות של מערכות הבינה המלאכותית באמצעות שילוב ידע ארגוני ייחודי
- להעריך את איקות מערכות GAR באמצעות מדדים כמותיים
- לקבל החלטות מושכלות בנוגע לסטרטגיות חיתוך מסמכים, מודלי gniddebmE ובסיסי נתונים וקטוריים

1.7 המהפהה השקתה בידע הארגוני

בשנת 3202, מנהלת משאבי אנוש בחברת הייטק בינוונית עמדה בפני אתגר מוכר: מאות עובדים פנו מדי יום בשאלות על מדיניות החברה - ימי חופשה, הילכי אישור הוצאות, זכויות הורים, ניהול מרוחק. התשובות היו קבועות במאגר עצום של מסמכים, מצגות ונוהלים שהצטברו לאורך שנים. הפתרון המסורתני - העסקת צוות תמיכה גדול או בניהת מערכת שאלות ותשובות סטטיסטית - היה יקר ולא יעיל.

از היא פנתה לפתרון חדש: בניהת מערכת GAR שמאפשרת לעובדים לשאול שאלות בשפה טבעית ולקבל תשובות מדויקות, מבוססות על המדיניות האמיתית של החברה. תוך שבועיים, המערכת ענתה על ממלה מ-08% מהפניות באופן עצמאי, תוך חיסכון של שעوت שעבודה שבועיות ושיפור משמעותם בשבועות רצון העובדים. זו המהפהה השקתה של GAR - הטכנולוגיה שmagshret בין הכוח הגנרטיבי של מודלי שפה גדולים לבין הידע הייחודי והמתעדכן של הארגון [7], [8].

2.7 מהו GAR? השילוב שמשנה הכל

2.7.1 הבעייה: ידע סטטי בעולם דיגיטלי

מודלי שפה גדולים כמו TPG-4 או edualC הם כלים מורשיים, אך הם סובבים מגבלה משמעותית: הם "קפואים בזמן". הידע שלהם נקבע במהלך האימון, ולא מתעדכן באופן אוטומטי. כאשר אתם

שואלים את CTPGtah של מדיניות החופשות בחברה שלכם, או על המפרט הטכני של המוצר החדש שהשיקתם בחודש שעבר, המודל פשוט לא ידוע. הוא לא יכול לדעת. הפתרון המסורי - לאמן מחדש את המודל על הנתונים שלכם - הוא לא מעשי. זה יקר, איטי, ודורש מומחיות טכנית עמוקה. ומה קורה כשהמדיניות משתנה? תאמנו מחדש שוב?

2.2.7 הפתרון: אחזור קודם יצירה

פותר את הבעיה הזה בגישה אלגנטית: במקום לשנות את GAR (noitareneG detnemguA-laveirteR) המודל, אנחנו מנסים את הקלט שלו. התהליך מורכב משני שלבים:

שלב א': אחזור (laveirteR) - כאשר משתמש שואל שאלה, המערכת מחפשת במאגר המידע הארגוני את המسمכים הרלוונטיים ביותר. החיפוש הוא חכם - לא רק לפי מילוט מפתח, אלא לפי משמעות סמנטית.

שלב ב': יצירה (noitareneG) - המسمכים שאוחזו מועברים למודל השפה יחד עם השאלה המקורי, והמודל יוצר תשובה מבוססת על הקשר המורחב הזה. הרעיון פשוט אך חזק: אנחנו לא ממלדים את המודל את המידע הארגוני, אנחנו מספקים לו את המידע הזה בזמן אמיתי, בבדיקה כשהוא צריך אותו [9].

3.2.7 למה זה עובד כל כך טוב?

השילוב בין אחזור ויצירה מנצח את החזוקות של שתי טכנולוגיות:

-- **מערכות אחזור מיידן** מצוינות במציאת מסמכים רלוונטיים במאגרים גדולים. הן מהירות, יעילות, וניתנות לעדכון מיידי.

-- **מודלי שפה גדולים** מצוינים בהבנת הקשר, סינטזה של מידע וייצור תשבות קוהרנטיות בשפה טבעית.

כאשר אתם משלבים אותם, אתם מקבלים מערכת שմשלבת את המידע העדכני של הארגון עם יכולות ההבנה והתקשרות של הבינה המלאכותית. זו לא רק שאלות ותשובות - זו הבנה אמיתית של הקשר ויכולת לספק תשבות מותאמות אישית.

3.7 הארכיטקטורה: מסע הנתונים דרך המערכת

הבנייה ארכיטקטורת GAR היא קריטית לתכנון והטמעה נכונה. בוואו נעקوب אחר מסע של נתון בודד - מסמך מדיניות - מהרגע שהוא נכתב ועד שהוא משמש לענות על שאלת עובד.

1.3.7 שלב 1: הכתת מסמכים - Cgniknuh

המסמך המקורי - נניח מדיניות חופשות בת 51 עמודים - הוא ארוך מדי בשbill להעברו אותו כולםodel השפה עם כל שאלה. זה לא רק בעקבות טוקנים (וכסף), אלא גם גורם ל"רעש" שמקשה על המודול למצוא את המידע הרלוונטי.

לכן, אנחנו מחלקים את המסמך לקטעים קטנים יותר - "sknuhC". אבל איך?

אסטרטגיית החלוקה הפושאה: גודל קבוע

הגישה הבסיסית ביותר היא לחלק לפי מספר מילים או תווים קבוע. למשל, כל Cgniknuh יכול 500 מילים.

יתרונות:

-- פשוט לIMPLEMENTATION

-- צפוי ועקביו

-- קל ליחסוב עלויות

חסרונות:

-- עלול לחתוֹך בamuצע משפט או רעיון

-- מתעלם מבניה המסמך

-- עלול להפריד בין מידע הקשור

אסטרטגיית החלוקה המבנית: לפי סעיפים

גישה חכמה יותר היא לחלק לפי המבנה הטבעי של המסמך - כותרות, פסקאות, רשימות.

תוֹיתַנְשׁ תוֹשְׁפּוֹחַ תוֹיִנְיִדָּמְ #

תוֹאַכְזַׁ #

...הנשב השפוח ימי 22-ל יאכז דבוע לכ

הריבץ #

... ישודוח לפואב מירבצן השפוחה ימי

כל סעיף משנה הופך ל-C sknuh נפרד, שומר על ההקשר השלם שלו.
 יתרונות:

-- שומר על שלמות רעיון

-- מכבד את כוונת המחבר

-- מייצר C sknuh בעלי משמעות

חסרונות:

-- sknuh בגדלים משתנים

-- סעיפים ארוכים מאוד עדין בעייתיים

-- דורש ניתוח מבנה המסמך

אסטרטגיית החלוקה החכמה: S citnamegniknuh

הגישה המתקדמת ביותר משתמשת לבינה מלאכותית כדי לאזות גבולות טבעיות בין רעיונות.
התהילך:

1. חלק את המסמך למשפטים

2. חשב E gniddebm לכל משפט

3. מצא נקודות שבחן הדמיון הסמנטי יורך משמעותית

4. חתוֹך שם

זו הדרך הטובה ביותר לשמר על שלמות רעיון, אבל היא גם cocci מורכבת ויקרה.

2.3.7 שלב 2: יצרת Esgniddebm - הפיכת טקסט למספרים

עכשו שיש לנו Csknuh, צריך להפוך אותו לפורמט שמחשב יכול לעבוד איתו ביעילות - וקטוריים במרחב רב-ממדי.

מהו gniddebm?

gneiddebm הוא יציג מתמטי של משמעות. במקומות לראות את המשפט "העובד זכאי לחופשה שנתית" כרץ' של תווים, אנחנו מייצגים אותו כוקטור של מאות או אלפי מספרים. הקסם הוא שמשפטים בעלי משמעות דומה יקבלו sgnddebm דומים - קרובים למרחב הווקטורי. המשפט "זכאות לימי מנוחה" יהיה קרוב למשפט על חופשה שנתית, למרות שאין בו אותן מילים בדיק.

מודלי gniddebm - השוואה

טבלה 1.7 מציגה השוואה בין מודלי ה-gniddebm המובילים בשוק [2], [21]:

Model	Dimensions	Performance	Cost
text-embedding-3-small	1536	Good	Low
text-embedding-3-large	3072	Excellent	Medium
NV-Embed-v2	4096	Excellent	Medium
BGE-M3	1024	Good	Free (Self-hosted)

טבלה 1.7: השוואת מודלי gniddebm

איך לבחור מודל gniddebm?
שכלו את השאלות הבאות:

-- **שפה:** האם המסמכים בעברית? לא כל המודלים תומכים היטב בעברית. מודל 3-M3-BGE [2] הוא רב-לשוני ועובד טוב עם עברית.

-- **תחום:** האם המסמכים טכניים מאוד? מודלים שאומנו על תחומיים ספציפיים יעבדו טוב יותר.

-- **עלות vs ביצועים:** מודלים גדולים יותר יקרים יותר לאחסן ולהפץ, אך מודדים יותר.

-- **פרטיות:** מודלים detsoh-fleS כמו 3M-EGB שומרים על המידע אצלם.

לדוגמה, עבור מאגר מסכמי RH בעברית, detsoh-fleS 3M-EGB עשוי להיות בחירה מצוינת. עבור מאגר טכני באנגלית, egral-3-gniddebm-txet יתן תוצאות מעולות.

3.3.7 שלב 3: אחסון במאגר וקטורי - esabataD rotceV

gneiddebm מאוחסנים במאגר נתונים מיוחד שמותאם לחיפוש וקטורי מהיר.
למה לא SQS רגיל?

מסד נתונים יחסית מסורתית גבוהה ליחסים מדויקים: "מצאת את כל העובדים שנשכחו ב-3202". אבל הוא איטי מאד לחיפושים סמנטיים: "מצאת את המסמכים הדומים ביותר למשפט זהה". esabataD rotceV מותאם במיוחד לשאילתה: "מי ה-K-nearest neighbors הקרובים ביותר לווקטור זהה?" - בדיק מה שאנו צריכים ל-GAR.

השווות מאגרי נתונים וקטוריים
esabataD - המנהל בענן [11]
esabataD הוא פתרון SaaS מנהל במלואו.
 יתרונות:

-- אפס תחזקה - הכל מנהל

-- סקייל אוטומטי

-- ביצועים מצוינים

-- בטוח וגבויים אוטומטיים

חסרונות:

-- עלות גבוהה בנפחים גדולים

-- הנזונים בענן חיצוני

-- תלות בספק

מתי להשתמש: כאשר אתם רוצים להתחילה מהר, אין לכם תשתיית, ואתם מוכנים לשלם עבור נוחות.

[4] amorphC - הפשט והמקומי amorphC

הוא מאגר קוד פתוח שקל להתקנה ושימוש. יתרונות:

-- קל מאד להתחילה - פחות מ-01 שורות קוד

-- ללא עלות (detsoh-fles)

-- מלא שליטה על הנזונים

-- טוב לפיתוח COP

חסרונות:

-- ביצועים מוגבלים בנפחים גדולים

-- אין תכונות esirpretnE מובנות

-- דורש תחזקה עצמית

מתי להשתמש: COP, פרויקטים קטנים, או כשאתם רוצים שליטה מלאה ואין לכם תקציב.

[31] etaivaeW - האיזון etaivaeW

הוא קוד פתוח עם אופציה למנהלה. יתרונות:

-- גמיש - detsoh-fles או duolc

-- ביצועים טובים גם בסקייל גדול

-- תכונות חיפוש מתקדמות

-- קהילה פעילה

חסרונות:

-- עקומת למידה תלולה יותר

-- דורש תכנון אדריכלי

מתי להשתמש: פרויקטים ברמת esirpretnE שדריכים גמישות, או כשאתם עוסקים COP-לייצור.

lavirteR - שלב 4: חיפוש 4.3.7

כעת מגע הרגע האמייתי. משתמש שואל: "כמה ימי חופשה מגיעים לי?"
התהיליך:

1. השאלת עוברת דרך אותו מודל gniddebmE שיצר את ה-C sknuh.
 2. נוצר gniddebmE של השאלה.
 3. מփש את ה-C sknuh הקרובים ביותר (בדרך כל esabataD rotceV).
 4. ה-C sknuh מוחזרים עם ציון דמיון.

הHIPPOSH הנפוץ ביותר הוא ytiralimiS enisoC - מדידת האזווית בין שני וקטורים. hcraeS ytiralimiS - איך זה באמת עובד?

$$\text{Cosine Similarity} = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

ציוויליזציה של 1 = זהים להלוטין, ציוויליזציה של 0 = שונים להלוטין.
כמה sknuhC להחזיר?
זהו ffo-edart קלסי:

- **מעט sknuhC (1-3):** מהיר, זול, אבל עלול להחמיר מידע חשוב
- **רובה sknuhC (01+):** מكيف, אבל יקר, איטי, ועלול להציג את המודל ב"רעש"
- בפועל, tops teews הם sknuhC 5-3 לרוב היישומים.

שלב 5: יצירת התשובה - G 5.3.7

עת אנהנו מרכיבים את ה- P_{tmor} הסופי למודל השפה:

System Prompt:

תולאש לע הנע . שונא יבашם תויינידמל החמומ רזוע התא
תאצ רמא , קייפסח אל עדימה מא . קפוש עדימה לע קר ספטהוב

Context:

[...השופחן תואכז לע פיעס :1
[...סיממי תרייבצ לע פיעס :2
[...סידחוים סיאנת לע פיעס :3

User Question:

? יְלִי מַעֲמִיכָם הַשְׁפּוֹת יְמִי הַמִּכְ

המודל מקבל את כל ההקשר הזה ויוצר תשובה מבוססת עובדות.

מדידת הצלחה: 4.7 scirteM noitaulavE

מערכת GAR יכולה להראות מרשימה, אבל איך אתם באמת יודעים שהוא טוב? מנהלים, אתם צריכים מדריכים כמותיים לקבלת החלטות [3], [6].

1.4.7 IlaceR - האם מצאנו את כל הרלוונטי?

מודד: מתוקן כל המסמכים הרלוונטיים שקיים במאגר, כמה באמת אוחצרו?

$$\text{Recall} = \frac{\text{ורזוחאש סיטונולר מיכמסם}}{\text{רנאט מיטנולר מיכמסם כ''הס}}$$

דוגמה:

במאגר יש 01 מסמכים שעונים על השאלה "מהי מדיניות העבודה מהבית?". המערכתacha אחזהה 5 מהם.

$$\text{Recall} = \frac{5}{10} = 0.5 = 50\%$$

נמוך אומר שאנו מפספסים מידע חשוב. זה בעיתי במיוחד בתחום רגולטוריים (משפט, רפואי, פיננסים) שבהם החמתה מידע יכולה להיות מסוכנת.

איך לשפר IlaceR?

-- הגדיל את מספר hnC-sknuhs שמוחזרים

-- שפר את איקות hnC-sknuhs (מודל טוב יותר)

-- בדוק את אסטרטגיית hnC-sknuhs - אולי hnC-sknuhs גדולים מדי או קטנים מדי

2.4.7 noisicerP - האם שמצאנו באמת רלוונטי?

מודד: מתוקן כל המסמכים שאוחצרו, כמה באמת רלוונטיים?

$$\text{Precision} = \frac{\text{ורזוחאש סיטונולר מיכמסם}}{\text{ורזוחאש מיכמסם כ''הס}}$$

דוגמה:

המערכת אחזהה 7 מסמכים. 5 מהם באמת רלוונטיים, 1-2 לא.

$$\text{Precision} = \frac{5}{7} \approx 0.71 = 71\%$$

נמוך אומר שאנו מציפים את המודל במידע לא רלוונטי, מה שעלול להוביל לתשובות שגויות או מבלבלות.

איך לשפר noisicerP?

-- הקטן את מספר hnC-sknuhs שמוחזרים

-- הגבה את סף הדמיון המינימלי

-- שפר את איקות המסמכים המקוריים (הסדר setacilpud, עדכן מידע ישן)

3.4.7 erocS 1F - האיזון המושלם

לעתים קרובות יש ffo-edart בין llaceR noisicerP ל-llaceR. אם תחזירו הרבה מסמכים, llaceR יעלה אבל noisicerP ירד. אם תחזירו מעט, ההפק 1F erocS מאזן בין שני המדרדים:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

דוגמה:
עם 17% של llaceR ו- 50% של noisicerP

$$F1 = 2 \times \frac{0.71 \times 0.50}{0.71 + 0.50} = 2 \times \frac{0.355}{1.21} \approx 0.587 = 58.7\%$$

1F erocS משמש בדרך כלל כמדד היחיד לאופטימיזציה, אבל זכרו: לעיתים אתם אכן רוצים להעדיין llaceR על noisicerP או להפץ, בהתאם למקרה השימוש.
מתי להעדיין llaceR?

- מערכות משפטיות - אסור להחמיר תקדים
- מערכות רפואיות - אסור להחמיר אזהרות
- תמיכת לקוחות - עדיף לתת יותר מידע מהחמיר
- מתי להעדיין noisicerP?**
- דוחות לניהול - רק מידע מדויק ומموקד
- מערכות המלצה - טוב יותר להמליט מעט מאשר להציג
- knuhC snoitacilppa evitisnes-tsoC - כל עולה כמספר

4.4.7 מדדים אינטיטיים נוספים

מעבר למדדים הכמותיים, שקלו גם:
 ecnaveleR rewsnA - האם התשובה הסופית עונה על השאלה?
 ssenlufhtiaf - האם התשובה נשארת נאמנה למסמכים המקורי?
 ycnetaL - כמה זמן לוקח לקבל תשובה?
 yreuQ rep tsoC - כמה עולה כל שאלתה?

5.7 דוגמאות מעשיות: GAR בפועל

1.5.7 דוגמה 1: GAR על מאגר מדיניות RH

האתגר:

חברת טכנולוגיה בינלאומית עם 005 עובדים ב-5 מדינות. כל מדינה עם חוקי עבודה שונים, מדיניות חופשotta שונה, הטבות שונות. מאגר של 002+ מסמכים מדיניות, הנחיות, שאלות נפוצות. צוות RH מקבל ממוצע של 05 פניות ביום. זמן תגובה ממוצע: 4 שעות. שביעות רצון עובדים: בינונית.

הפתרון:

בנinit מערכת GAR ייעודית למאגר ה-RH:
שלב 1 - הכנות הנתונים:

-- איסוף כל המסמכים לתיקיה מרכזית

-- המרה של FDP/XCOD/COD לפורמט טקסט

-- תיוג כל מסמך לפי מדינה וושא

שלב 2 - gniknuhC :

החליטו על gniknuhc מבני: כל סעיף משנה במסמך הופך ל-knuhC נפרד. כך, סעיף "זכאות לחופשה - ישראל" הוא knuhC אחד בלבד. כל knuhC מקבל :atadateM

{

"text": "...השפוח ימי 22-ל יאכז לארшиб דבוע",
"country": "Israel",
"topic": "vacation_policy",
"last_updated": "2024-01-15"

}

שלב 3 - gniddebmE ואחסון:

השתמשו ב-llams-3-gniddebmme-txet-enoceniP (מאזן טוב בין עלות לביצועים). אחסנו ב- (נבחר בגלל הנוחות והביצועים, העלות הייתה סבירה עבור 000 מסמכים).

שלב 4 - laveirteR מותאם:

כאשר עובד שואל שאלה, המערכת:

1. מזהה את מדינת העובד (מתוך פרטי המשתמש)

2. מבצעת חיפוש עם סינון: "learnsI" = yrtnuoc

3. מחזירה sknuhC 3 hei דומים מישראל בלבד

זה מונע הבלבול עם מדינות אחרות.

התוצאות:

-- 57% מהפניות נענות אוטומטית ללא התערבות אנושית

-- זמן תגובה ממוצע ירד מ-4 שעות ל-0.3 שעות

-- שביעות רצון עובדים עלתה ל-5/5.4

-- צוות RH חוסך 0.2 שעות שבועית, מושקעות בתמיכה מורכבת יותר

--rocS 1F של המערכת: 28%

2.5.7 דוגמה 2: GAR על תיעוד טכני של מוצרים

האתגר:

חברת SaaS עם 51 מוצרים שונים. תיעוד טכני עצום: מדריכי משתמש, IPA, noitatnemucod seton esaeler, gnitoohselbuort. מדריכי החיפוש אחר מידע. לקוחות מתוסכלים מזמן ההמתנה.

הפתרון:

מערכת GAR פנימית לצוות התמיכה + tobtahc ללקוחות.

gniknuhC :yetartS

תיעוד טכני מובנה היטב - השתמשו ב-sredaeh nwodkraM sknuhC sredaeh כגבילות

```

# API Reference
## Authentication
### API Key
Each request must include...
[Chunk אזן]

```

OAuth 2.0
For applications that need...
[Chunk אזן]

:gniddebmE
egral-3-gniddebme-txet - מסמכים טכניים דורשים דיק נבוה.
:esabataD rotceV
detsoh-fles etaivaeW - נפח גדול (sknuhC +000,05), צרכיים ביצועים וגמישות.
:laveirteR
hcraeS dirbyH - שילוב של:

(citnames) hcraes rotceV --

(sehctam tcaxe) hcraes drowyeK --

למשל, חיפוש "104 rorre IPA" ימצא גם:

"104 rorre" sknuhC --

(citnames) sknuhC --

[01] :gniknar-eR

אחרי אחוזר ראשוני של 01 sknuhC, מודל נפרד (redocnE-ssorC) מדרג מחדש ומוחזר את 3 הטובים ביותר.
התוצאות:

-- צוות תמייה פוטר בעיות 04 % מהר יותר

-- toktahc לקוחות פוטר 06 % מהפניות באופן אוטומטי

-- ירידה של 03 % בזמן המתנה ממוצע

-- :noisicerP ,%98 :llaceR --

3.5.7 דוגמה 3: GAR על היסטורית תמיכת לקוחות

האתגר:

מועד שירות לקוחות עם 5 שנים היסטוריה - 000,001+ שיחות, מיילים, stekcit. מלא ב"ازב"
- פתרונות לביעות נדרות, דוגמאות לטיפול מוצלח במצבים מורכבים.
אבל הידע זהה לא נגיש. כל נציג "מציא את הגלגל מחדש".

הפתרון:

GAR על כל ההיסטוריה התמיכה.

:noitaraperP ataD

אתגר גדול - הנתונים מגוונים:

MRC-stekciT -- מובנים מ-

-- מיילים חופשיים

-- תמלולי שיחות

-- הערות פנימיות

פיתחו enilepip "יעודי":

1. ניקוי נתונים - הסרת מידע אישי (RPDG)

2. זהויoituloseR - רק stekcit שנסגרו בהצלחה

3. סיכום - כל tekcit ארוך סוכם לכדי 200 מילים

:gniknuhC

כל knuhC = tekcit אחד (אחורי הסיכום).

MdateM: atadat

{

"problem": "עובד שחל רבעתnal זילצט אל",
"resolution": "זקין + המסיס סופיא cache",
"product": "Mobile App",
"resolution_time": "15 minutes",
"customer_satisfaction": 5

}

:laveirteR חכם:

כאשר נציג פותח tekcit חדש:

1. המערכת מזהה את הבעיה

2. מחפשת דומים stekcit שנפתחו

3. מסנת רק פתרונות עם noitcafsitas גובה

4. מציעה לנציג: "בעיות דומות נפתרו בעבר כך..."

התוצאות:

-- זמן פתרון ממוצע ירד ב-52%

-- שבעיות רצון לקוחות עלה ב-51%

-- נציגים חדשים יעילים פי 2 מהר יותר

-- הפחתת snoitalacse ב-50%

6.7 אתגרים ופתרונות: מה שאף אחד לא מספר לכם

GAR נשמע מדהים בתיאוריה, אבל בפועל יש אתגרים. בואו נדבר על האתגרים האמתיים והפתרונות המעשיים.

1.6.7 אתגר 1: "זה לא עובד בעברית"

רובה המודלים מאומנים בעיקר על אנגלית. עברית? לא תמיד טוב.
הסימפטומים:

-- sknuhC בעברית מקבלים ציוני דמיון נמוכים

-- שאלות בעברית מוצאות תשובות באנגלית

-- ביצועים ירודים לעומת אנגלית

פתרונות:

1. **השתמשו במודל רב-לשוני:** 3M-EGB, 5E-laugnilitluM, gniddebmE - מאומנים על عشرות שפות כולל עברית

2. **תרגומם לאנגלית:** תרגמו את המסמכים לאנגלית לפניgniddebmE (יקר, אבל עיל)

3. **אמנו את מודל ה-gniddebmE:** אמנו את מודל ה-gniddebmE על קורפוס עברית ספציפי בתחום שלהם

2.6.7 אתגר 2: "המערכת זהה"

לפעמים המודל מחזיר תשובה שנראית מהימנה, אבל לא מבוססת על המסמכים.
למה זה קורה:

-- ה-sknuhC שאחזרו רלוונטיים חלקיים בלבד

-- המודל "מצחיא" מידע מהידע הכללי שלו

-- ה-tpmorP לא מספיק נוקשה

פתרונות:

1. **tpmorP חמוץ:**

1. **tpmorP חמוץ:** הכריחו את המודל לcztt:

שروعם בرمאה, מיכטט מיריק אל עדימה מא

"מייניזה מיכטט זכ לע עדימ ייל זיא"

ילאך עדיב שמתחעל זל רוסא.

2. **noitatiC:** הכריחו את המודל לcztt:

זוקל אווה זמסם הזיאם זייצ, הבושתב טפשם לכל

X דומע, זמסמה מש[ב בותכש יפכ" :טמרוף ..."

3. **pool noitadilaV:** בדקו את התשובה מול המסמכים:

-- תננו למודל נפרד לבדוק: "האם התשובה נתמכת בmsemcics?"

-- אם לא - דגלו או דחו

3.6.7 אתגר 3: "המידע מיושן"

מסמך עודכן, אבל המערכת עדין מחזירה את הגרסה הישנה.
למה זה קורה:

- לא עדכנתם את ה-V-rotceD esabataD
- יש setacilpud - גרסה ישנה וחדשה
- לא ברור למערכת אם גרסה עדכנית

פתרונות:

:gninoisreV .1. knuhC מקבל noisrev ו-pmatsemit-

```
{  
    "text": "...",  
    "document": "vacation_policy",  
    "version": "2.3",  
    "last_updated": "2024-03-15"  
}
```

.2. :A-otuA-ferserh: תהליך אוטומטי שבודק שינויים:

- rotinoM מסמכי תיקיות
- sknuhC חישנים מושתנה, מחק את ה-V-chidshim יוצר sknuhC
- esabataD rotceV עדכן את ה-V-

.3. :M-ateadat-lif-retni-gn: בחיפוש, העדיף תמיד את הגרסה החדשה:

```
# Retrieval  
filter = {"last_updated": {"$gte": "2024-01-01"} }
```

4.6.7 אתגר 4: "זה מאד יקר"

עם מאגר גדול, הבעיות יכולות להיות משמעותיות:

- gniddebmE מילוני sknuhC
- rotceV esabataD אחסון ושאלות
- MLL sknuhC yreuq כולל כל

פתרונות:

.1. :C-ahcina-gnihc:

- נפוצות לשאלות שמיירת תשובה
- מ-ehcaC החזר נשאלת, דומה שאלה אם

.2 :sgniddebmE rellamS

-- במקום snoisnemid 2703, השתמש snoisnemid 6351 או פחות snoisnemid sgndidebmE akhsoyrtam --

.3 :hcaorppa dirbyH

-- שאלות פשוטות - חיפוש drowyek בלבד (אול)

-- שאלות מורכבות - GAR מלא (יקר)

.4 :detsoh-fleS

-- gniddebmE 3M-EGB על שרת שלכם - ללא עלות

-- esabataD tnardQ/amorphC - ללא עלות

-- MLL lacol 3 amalL - ללא עלות

7.7 תכנו תהליכי עדכון ידע ב-GAR

מערכת GAR היא ארגניזם חי. הידע הארגוני משתנה כל הזמן - מדיניות מתעדכנת, מוצרים משתנים, נוהלים משתפים. איך מתחזקים את המערכת?

1.7.7 אסטרטגיות עדכון

.1. בניית מחדש מלאה dlilubeR lluF

כל שבוע/חודש, מחק הכל ובניו מחדש.
 יתרונות:

-- פשוט

-- מבטיח ycnetsisnoc

-- אין setacilpud

חסרונות:

-- יקר (gniddebmE חדש של הכל)

-- (stnemnorivne emitnwod) או צורץ בשני

-- בזבוז על מסמכים שלא השתנו

מתי להשתמש: מאגרים קטנים (> 000,01 scod), שינויים נדירים.

.2. עדכון מצבה etadpU latnemercnI

עקבות אחורי שינויים ועדכון רק מה שצריך.
תהליכי:

(etad deifidom) segnahc elif rotinoM .1

2. זהה מסמכים שהשתנו

3. מחק רק את ה-sknuhC של מסמכים אלה

4. צור E sgniddebmE חדש רק להם

5. הוסף ל-rotceV esabataD

יתרונות:

-- עיל - עדכון רק מה שצרי

-- מהיר

-- זול

חסרונות:

-- מרכיב יותר

-- צריך gnikcart

-- עלול להחמייך שינויים

מתי להשתמש: מאגרים גדולים, שינויים תכופים.

3. etadpU emit-laeR - עדכון בזמן אמיתי

כל פעולה על מסמך מייד מעדכנת את ה-GAR.

תהליך:

renetsil tnevE/koohbeW -- על מערכת הקבצים

-- מסמך נוסף ddA + debmE --

-- מסמך עודכן wen ddA + debmE + dlo eteleD --

-- מסמך נמחק BD rotceV morf eteleD --

יתרונות:

-- תמיד עדכני

-- אין yaled

חסרונות:

-- הכי מרכיב

-- עלול להעמיס על המערכת

-- צריך erutcurtsarfni חזקה

מתי להשתמש: כאשר emit-laeR קרייטי (תמייה בזמן אמיתי, מערכות ייצור).

2.7.7 עדכון מומלץ enilepiP

לארגון טיפוסי, הנה enilepip מאוזן:

.1. כל לילה, בדוק שינויים ועדכן :latnemercnI ylthgiN

.2. פעם בשבוע, מלא dluber lluF ylkeeW (בתיוחות)

.3. אפשרות לעדכון מיידי במקרה חירום :reggitT launaM

```
# Pseudo-code
schedule.every().day.at("02:00").do(incremental_update)
schedule.every().sunday.at("03:00").do(full_rebuild)

def incremental_update():
    changed_files = get_files_modified_since_last_run()
    for file in changed_files:
        old_chunks = get_chunks_for_file(file)
        delete_from_vector_db(old_chunks)

        new_chunks = chunk_document(file)
        embeddings = embed_chunks(new_chunks)
        add_to_vector_db(new_chunks, embeddings)

    log_update(changed_files)
```

8.7. *בנייה תרבות נתונים: המפתח להצלחת GAR*

המכשול הנDSL ביוטר להצלחת GAR הוא לא טכנולוגי - הוא ארגוני.

1.8.7. *aicot נטונים היא הכל*

GAR טוב כמו הנתונים שהוא מבוסט עליהם. "tuo egabrag ,ni egabraG" - אם המסמכים הארגוניים מבולגניים, מיוושנים, או סותרים, GAR לא יכול אתכם. **בעיות נפוצות:**

-- 5 גרסאות של אותה מדיניות, לא ברור איזו עדכנית

-- מידע מישן: מסמכים מ-5102 ש כבר לא רלוונטיים

-- פורמטים מבולגניים: FDP סרוק שלא ניתן לחילוץ טקסט

-- חוסר עקביות: מחלקות שונות קוראות לאותו דבר בשמות שונים

פתרונות: ecnanrevoG ataD
לפני שאתם בונים GAR, השקיעו בינוי וארגון:

1. **סקר מגאר:** מה יש לנו בכלל?

2. **ניקוי:** מחיקת ישן, איחוד setacilpud
3. **סטנדרטיזציה:** פורמט אחיד, מינוח אחיד
4. Opihsrenw: כל מסמך מקבל אחראי לעדכון
5. **תהליכיים:** איך מוסיפים/معدכנים/מוחקים מסמכים

2.8.7 שינוי תרבותי

- GAR מצליח כאשר הארגון מאמץ "erutluC tsriF-ataD":
- **תיעוד הוא אחריות:** כל מחלקה חייבת לתעד את הידע שלה
 - **עדכניות היא קריטית:** מדיניות שונה יותר יותר מאי-מדיניות
 - **שકיפות:** מידע לא חסוי צריך להיות נגיש לכלום

9.7 העתיד: לאן הולך GAR?

GAR התפתח מאד בשנים האחרונות, והוא ממשיך להתקדם במהירות מScheduler.

1.9.7 טrndים מתוערים

GAR [1] - סוכנים שלחלייטים עצמם: citnegA

- האם צריך GAR או לא
- מאייה מאגר לשלוּף
- כמה sknuhC לאחזר
- האם צריך חיפוש נושא

GAR ladomitluM - לא רק טקסט:

- אחזר תМОנות, דיאגרמות
- חיפוש בוידאו
- שילוב אודיו

GAR hparG [5] - GAR המבוסס על גרפי ידע:

- במקומות sknuhC בודדים, גוף של יחסים
- הבנת קשרים מורכבים

-- הסקת מסקנות חדשות

GAR detaredeF - חיפוש על פני מאגרים מרוביים:

- חלק מהידע אצלם, חלק אצל שותפים
- שמירה על פרטיות
- אחזר מבואר

2.9.7 האתגרים הבאים

הערכת אוטומטית - כיום הערכת GAR דורשת עבודה ידנית. בעtid:

-- מדדים אוטומטיים בזמן אמיתי

-- זיהוי בעיות לפני שימושים רואים עצמם

-- שיפור מתמיד (gninraeL suounitnoC)

-- GAR שמתאים עצמו לכל משתמש: noitazilanosreP

-- רמת מומחיות שונה = sknuhC שונים

-- היסטוריה אישית משפיעה על laveirteR

-- סגנון תשובה מותאם

-- GAR יקר. העtid: noitazimitpO tsoC

-- מודלים קטנים ויעילים יותר

-- sknuhC חכם ומדויק יותר = פחות laveirteR

-- gnihcaC אגרסיבי

01.7 סיכום: GAR כמקור יתרון תחרותי

GAR הוא הרבה יותר טכנולוגיה טכנית - הוא גשר בין הידע הארגוני הייחודי שלכם לבין הכוח של בינה מלאכותית גנרטיבית. הארגונים שמצילים להטמע GAR ביעילות מקבלים יתרונות משמעותיים:

-- **נגישות ידע:** עובדים מקבלים תשובות מהירות ומדויקות

-- **עקיבות:** כולם מקבלים את אותו המידע, מקור אחד

-- **יעילות:** חישכון בזמן חיפוש ושאילת שאלות

-- **סקלבilities:** מערכת אחת משרתת אלפי משתמשים

-- **שיפור מתמיד:** ככל שמוסיפים מידע, המערכת משתפרת

אבל זכרו: GAR הוא כלי, לא פתרון קסם. ההצלחה תלולה בתכנון נכון, נתונים איקוטיים, הטמעה מושכלת, ותרבות ארגונית תומכת. בפרק הבא עמוק באמנות כתיבת stpmorP אפקטיבים - המימוניות שתקבע האם GAR שלכם יהיה טוב או מעולה.

11.7 תרגילים

1.11.7 תרגילים תיאורתיים

תרגיל 1: תוכנן מערכת GAR למסמכים בארגון שלך
בחר מקרה שימוש ספציפי בארגון שלך (למשל: מדיניות RH, תיעוד מוצר, נהלים תפעוליים) ותוכנן מערכת GAR מלאה:

1. זהה את מקורות הנתונים (סוגי מסמכים, מקום, פורמטים)
2. בחר yggetartS gniknuhC מותאמת והצדק
3. בחר ledoM gniddebmE והסביר למה
4. בחר esabataD rotceV והשווה לאלטרנטיבות
5. תוכנן תהליך עדכון
6. הגדר מדי הצלחה
7. הערך עליות (seireuQ ,egarotS ,gniddebmE)

תרגיל 2: בחר yggetartS gniknuhC מותאים
העבר כל אחד מסוגי המסמכים הבאים, המליך על yggetartS gniknuhC והסביר:

1. חוזים משפטיים - מסמכי FDP בני 05+ عمودים, מאוד מובנים (סעיפים, תת-סעיפים)
2. מיילים פנימיים - הודיעות קצרות עד בינוין, לא מובנות
3. מצגות שיוקיות - tnioPrewoP עם טקסט ותמונות
4. קוד תוכנה - קבצי nohtyP עם תיעוד enilni
5. פוסטים בפייסבוק פנימי - דיוונים עם שאלות ותשובות

תרגיל 3: השווה בין sesabataD rotceV ל课文
בhinaten התרחישים הבאים, בחר esabataD rotceV וסביר:
תרחיש א': סטארטאפ, COP למערכת תמיכת לקוחות, 1,000 מסמכים, תקציב מוגבל, צרכיים להציג תוצאות תוך שבועיים.
תרחיש ב': תאגיד בינלאומי, 5,000,005 מסמכים, דרישות RPDG מחמירות, תקציב ממשמעותי, אין מומחיות D.spOveD.
תרחיש ג': חברת ביוטה, 5,000 פוליסות, נתונים רגיסטרים, דרישת לשמירה מקומית, יש צוות IT חזק.
העבר כל תרחיש:

- המליך על etaivaeW/amorphC/enoceniP)
- הסביר את ההחלטה
- פרט יתרונות וחסרונות
- הערך עליות

תרגיל 4: תכנן תהליכי עדכון מידע ב-GAR
עבור מערכת GAR על מדיניות חבורה:

-- מדיניות מתעדכנת בממוצע פעם בחודש

-- כ-200 מסמכים במאגר

-- כ-50 משתמשים יומ-יומיים

-- קרייטי שהמידע יהיה עדכני (רגולציה)

תכנן:

1. אסטרטגיית עדכון (emit-laeR/latnemercnI/lluF)

2. תדריות עדכון

3. תהליכי וולד ציה - איך מודאים שהעדכון הצליח?

4. תוכנית kcablloR - מה קורה אם העדכון משבש?

5. איך מודיעים למשתמשים על שינויים?

תרגיל 5: בנה מדדי הצלחה למערכת GAR
עבור מערכת GAR לתミニכת לקוחות, הגדר:

1. 3 מדדי ביצועים טכניים (...ycnetaL, 1F, llaceR, noisicerP)

2. 3 מדדי ביצועים עסקיים (...sgnivaS tsoC, emiT noituloseR, TASC)

3. סך (dlohserhT) לכל מדד -מתי המערכת "מספיק טוביה"?

4. תוכנית מדידה - איך ומתי תמדדוו?

5. תוכנית שיפור - מה תעשו אם המדים לא מספקים?

2.11.7 תרגילי קוד - nohtyP

תרגיל 6: בניית GAR פשוט עם BDamorhC
בנה מערכת GAR בסיסית שמאפשרת:

-- טעינת מסמכים מתקינה

-- חיתוך אוטומטי לפי גודל קבוע

-- יצירת sgniddebmE עם IAneP

-- אחסון ב-BDamorhC

-- חיפוש וקבלת תשובה

תרגיל 7: הרכבת ביצועי GAR עם מדדים
בנה מערכת הרכבה:

-- טען tes של שאלות-תשובות

-- עברו כל שאלת, בצע laveirteR

-- חשב 1F ,llaceR ,noisicerP

-- חשב ycnetaL ממוצע

-- הצג דוח מסודר