מבני נתונים – מטלה 4

סמסטר סתיו 2026

תאריך הגשה: 25.1

מטרה

יישום ההבנה שלכם בטבלאות גיבוב לצורך מימוש מערכת פשוטה ויעילה של Retrieval-Augmented Generation (RAG) באמצעות Python. תרגיל זה ידגיש כיצד פונקציות גיבוב משמשות ביישומים בעולם האמיתי לאחזור יעיל של נתונים בעלי דמיון סמנטי.

רקע: טבלאות גיבוב פוגשות את RAG

מערכות RAG משפרות מודלי שפה על ידי מתן אפשרות לשלוף מידע רלוונטי ממאגר חיצוני במקום להסתמך אך ורק על ידע פנימי. בעוד שצינורות RAG קונבנציונליים משתמשים במאגרי מידע וקטוריים ומקודדים סמנטיים, בתרגיל זה תחקרו גישה מבוססת גיבוב. תעצבו פונקציית גיבוב משלכם שתקבץ שאילתות דומות סמנטית יחד, מה שיאפשר אחזור מהיר דרך טבלת גיבוב.

הערה: התיאור של *RAG* כאן הוא מינימלי בכוונה. אתם צפויים לחקור ולהרחיב עצמאית על המושג כדי להעמיק את הבנתכם.

מה תלמדו

- כיצד לעצב פונקציית גיבוב מותאמת אישית לשאילתות סמנטיות.
- כיצד לממש טבלת גיבוב לאחזור מהיר.
- כיצד גיבוב יכול להניע חלופות קלות משקל לארכיטקטורות RAG סטנדרטיות.

הוראות

שלב 1: הגדרת פונקציית גיבוב מותאמת אישית

עצבו פונקציית Python הממפה מחרוזת שאילתה למפתח בגודל קבוע (מספר שלם). פונקציה זו צריכה לשמר דמיון סמנטי מסוים – כלומר, שאילתות דומות צריכות להתגבב לאותם מפתחות או למפתחות קרובים.

דוגמה לשאילתות דומות:

1. "The lecturer and teaching assistant for the course are excellent!"
2. "The DS course team is outstanding"

רמז: ניתן לבצע טוקניזציה (*tokenization*) למחרוזת או להשתמש בתכניות תדירות מילים. הימנעו מהסתמכות על פונקציית ה-`hash()` המובנית של *Python*.

שלב 2: בניית טבלת הגיבוב

ממשו מבנה טבלת גיבוב משלכם (דמוי dictionary). השתמשו בשרשור (chaining) או במיעון פתוח לפתרון התנגשויות.

שלב 3: הכנסת פרומפטים לדוגמה

צרו מאגר נתונים קטן של זוגות פרומפט-תגובה (לפחות 10). בצעו גיבוב לפרומפטים ואחסנו אותם ב"דליים" (buckets) המתאימים.

שלב 4: מימוש שליפה (Retrieval)

כתבו פונקציית `retrieve(query)` אשר מבצעת גיבוב לשאילתה, מחפשת ב"דלי" המתאים ומחזירה את ההתאמה הטובה ביותר (או top-k) על בסיס מדד דמיון שתגדירו.

עליכם להכין את הבאים:

- `custom_hash(query: str) -> int`
- מימוש המחלקה `HashTable`
- פונקציות `insert` ו-`retrieve`
- דוח קצר (1-2 פסקאות) המשקף את ביצועי הפונקציה והטרייד-אופים שלה

הנחיות הגשה

הגישו את קוד ה-Python והדוח בתיקיית zip בשם `hash_rag_<ID>.zip`.

הערה: המשימה היא פתוחה. יצירתיות וקוד מתועד היטב הם המפתח. התמקדו בלמידה כיצד גיבוב יכול לשקף סמנטיקה של נתונים.