

## אופציה 3 לתרגיל בית מס' 5

# ניסוי מעבדה: התקנת Ollama RAG

מבנה כללי -- 4 תת-ניסוי

## 1 הקדמה

מטרה זו עוסקת בניתו ולימוד של התקנות מבוססות RAG ו-OLLAMA. שורת הניסויים המוצעת להלן מהוות מסגרת רעיהנית כללית, ואתם מוזמנים לפרש, לפתח ולחזור את הנושאים בכל דרך שתמצאו לנכון. עברו כל ניסוי עלייכם להגדיר שאלות מחקר, לבצע את הניסויים ולנתח את הממצאים, רצוי תוך הצגת ניתוח סטטיסטי ויאלי (באמצעות גרפים או טבלאות). מומלץ לחזור על כל ניסוי מספר פעמים כדי להבטיח תוקף סטטיסטי לתוצאה.

**משמעותו:** המשקנות שלכם אינן חייבות לחפות לחומר שהוצע בשיעור. אתם רשאים להגיע לתובנות עצמאיות, בלבד שתתמקו אותן היטב; במקרים אלו מומלץ להיעזר בסימוכין חיצוניים ולהציגו הסבר פפעריים שגיליתם. קחו את הניסויים למקום שמעניין אתכם ולכיוון החקירה האישית שלכם -- ההנחיות הללו נועדו לשמש כסייע מוחות' ואין בגדר הגדרות סגורות.

## 2 כלי עזר לשימוש בניסוי

ניסוי זה מיועד לחובבי ההתקנות הטכניות.  
הניסוי הזה מבוסס על הסרטון מיותר:

Ollama Course -- Build AI Apps Locally  
<https://www.youtube.com/watch?v=GWB9ApTPTv4&t=123s>

בנוסף, מצורפת חוברת הסבר שנכתבה על בסיס הסרטון.

## 3 תת-ניסוי 1: "Baseline" נכוון לפי הסרטון

מטרה: לשחזר 1:1 את מה שהמדריך הסרטון עשה -- כדי שייהי "קו ייחוס".

### 1.3 הסטודנט מתבקש

#### 1.1.3 להתקין ולהריץ

• Ollama (שרת מקומי כברירת המחדל על localhost:11434)

• מודל שפה בסיסי: למשל llama3.2 או llama3.2.3B (כפי שהומלץ)

• מודל שפה בסיסי: כמו pull :embedding :nomic-embed-text (דרך nomic-embed-text pull)

#### 2.1.3 לבנות RAG "קלאסי" כפי שמתוואר בספר/סרטון

שימוש ב-RecursiveCharacterTextSplitter עם chunk\_size≈1200, overlap≈300 :**Chunking** שהוצעו.

.model='nomic-embed-text' עם OllamaEmbeddings :**Embeddings**  
Chroma.from\_documents(..., embedding\_function=OllamaEmbeddings, persist\_directory=...) :**Vector Store**

"Answer the question based on the following context: ..."  
prompt ו- similarity\_search(query, k=3) :**Retrieval + Generation**

### 3.1.3 להפעיל סקריפט פיתון קצר שմבצע

- אינדוקס (חדר-פערמי) של מסמך קטן (למשל דף מדיניות פנימית קצר שמסופק לסטודנט)
- שאלת 2--3 שאלות פשוטות
- מדידת:
  - זמן אינדוקס (שניות)
  - זמן תשובה ממוצע
  - האם התשובה מדויקת (כן/לא, לפי פתרון ידוע)

### 4.1.3 הסטודנט חייב לוודא שהכל לפי ההוראות בסרטון

- אותו פקודות pull / create / run ollama embeddings
- אותה ספריית (Chroma) Vector Store ואותה אינטגרציה LangChain

## 2.3 תוצר: "Baseline Report" קצר

- תצורת מערכת (מודלים, ספריות, פורט, ספרייה persist)
- זמני ריצה
- איכות תשובות

## 4 תת-ניסוי 2: שינוי החלטה אחת -- ומה נשביר?

מטרה: להבין ריגישות ההתקנה והסתפק: שינוי קטן → השפעה טכנית/התנהגותית.  
הסטודנט בוחר לפחות 2 מהשינויים הבאים (על בסיס מה שהסרטון המליץ לא לעשות / לא עשה בפועל):

### 1.4 שינוי מודל ה-LLM

במקום llama3.2 → מודל קטן יותר (3B) או גדול יותר (8B).  
בדיקה:

- זמן תשובה (Latency)
- שימוש בזיכרון (אם אפשר למודול)
- איכות תשובה (אותן שאלות מניסוי 1)

## 2.4 שינוי מודל ה证实עה

במקום nomic-embed-text → מודל Embedding אחר (אם הסרטון/ספר מזכירים חלופה; או אפילו Embedding עני).  
בדיקה:

- איקות האחזור (אם עדין חוזרים אותם ?) Chunks
- שגיאות אפשריות בחיבור (כתובת שרת, API key, עיכוב רשת)

## 3.4 שינוי פרמטרים של Chunking

chunk\_size קטן מאוד (למשל 003) או גדול מאוד (0003), overlap שונה.

בדיקה:

- מספר ה-Chunks שנוצרו
- איקות התשובות: האם נוצר מצב "לא נמצא בהקשר" כי המידע נחתך/הוצף?

## 4.4 "שוכחים" להתמיד את ה-Vector Store

לא להשתמש ב-persist\_directory; כל ריצה בונה אינדקס מחדש.

בדיקה:

- זמן "אינדקס" בכל ריצה
- תחושת שימושיות/scalability (אינטואיציה על פרודקشن)

## 5.4 טבלת השוואة

הסטודנט מריץ מחדש את אותן שאלות מניסוי 1, ומתבקש למלא טבלה:

מדד	בעיות טכניות	האם התשובה מדויקת	מספר Chunks	זמן תשובה ממוצע	שינוי A	שינוי B

## 6.4 מטרת פדגוגית

לגרום לו להבין את המשמעות הטכנית של המלצות בסרטון: למה דוקא chunk\_size זהה, ומה Retrieval indexing, Chroma, nomic-embed-text, ומה להפריד בין-Chunks.

## 5 תת-ניסוי 3: מקומי מול חלופה חינמית בענין

מטרה: להמחיש את ההבדלים בין:

- "חריות חישובית מקומית" (Ollama) על הלפטופ שלך)
- שירות ענן חינמי/זול (למשל API ציבורי עם מגבלות)

הניסוי יכול להיות גם תיאורטי/סימולטיבי אם לא רוצים באמת לחבר API, אבל עדיף דוגמת קוד קונקרטי.

## **1.5 הסטודנט מtabק**

### **1.1.5 לשכפל את פייפליין ה-RAG**

להחליף את ה-LLM המקורי (llama3.2 דרך Ollama) ב-LLM עניינית (למשל מודל חינמי ב-Groq, Inference, Together, Vector Store, Embeddings, Chunking). להשאיר את כל שאר השלבים מקומיים:

### **2.1.5 להריץ את אותה שאלתה בבדיקה בשלושה "מצבים"**

- **מצב A:** הכל מקומי (Ollama LLama + nomic-embed-text + Chroma)
- **מצב B:** Embeddings מוקומיים, LLM בענן
- **מצב C:** Embeddings + LLM בענן (אם אפשר/רצו)

### **3.1.5 למדוד**

- Latency (כולל רשות)
- תלות ברשות (האם אפשר לרוץ offline?)
- תקלות אוטנטיקציה/Rate Limit
- איכות התשובה (צריכה להיות דומה)

## **2.5 המסקנות הרצויות**

- להבין את tradeoff: עלות/זמן/פרטיות/פשטות התקנה
- לראות שהארכיטקטורה עצמה (RAG) זהה, רק ה-"מנוע" מתחלף

## **6 תת-ניסוי 4: "סוגי RAG" ופתרונות מתקדמות**

מטרה: לקשור את התאוריה מהפרקים -- סוגי כשי RAG ואסטרטגיות Context -- למימוש טכני. הסטודנט יתנסה לפחות בשתי וריאציות:

### **Basic RAG vs Contextual Retrieval 1.6**

#### **Basic 1.1.6**

לפניה הטעינה, יוצרת "គוורת/תיאור הקשר" לכל מסמך (ע"י LLM), צירוף התיאור ל-Chunk ואז Embedding כל Chunk כמו שהוא.

#### **(Anthropic Contextual Retrieval 2.1.6**

לפני הטעינה, יוצרת "គוורת/תיאור הקשר" לכל מסמך (ע"י LLM), צירוף התיאור ל-Chunk ואז Embedding.

### **3.1.6 השוואת**

- האם שאלות "גבוליות" (למשל ניסוח מרומז) מצליחות יותר בשיטה השנייה?
- מדידה: כמה מתוך N שאלות חוזרות עם Chunk "נכון"

## **Basic Retrieval vs Reranking 2.6**

### **ריגיל Retrieval 1.2.6**

→ זריקה ישירה ל-LLM → similarity\_search(k=5)

### **Reranking 2.2.6**

→ שימוש ב-LLM ( מקומי או ענני ) לרה-דירוג לפי התאמה לשאלת →  
הכנסת רק טופ 3 ל-LLM לשאלת הסופית.

### **3.2.6 בדיקה**

האם יש פחות כשלים "Missed Top Rank / Not in Context" כפי שתוארו בפרק ה-anti-patterns

## **3.6 מטרת הניסוי**

להפוך את תאור ה-anti-patterns והסטרטגיות contextual ,rerank chunking (נכון, anti-patterns ממלל תיאורטי) לקוד מודיד.

## **7 דרישות מטה-פדגוגיות**

כדי להבטיח שהסטודנט באמת יזכה ב סרטון:

### **1.7 שאלות ספציפיות לסרטון**

בכל תת-ניסוי, יש " שאלה ספציפית לסרטון", לדוגמה:

- "אייזה מודל Embedding מומלץ הסרטון ומדוע?" → עליו זהה זאת מתוך הסרטון/תמלול
- "אייזה פורט השרת של Ollama ומתי צריך לשנות אותו?"
- "אייזה flags מופיעים בפקודה ollama create בדמו?"

### **2.7 צ'ק-лист סופי**

הסטודנט צריך לצרף:

- צילום מסך/פלט טרמינל של פקודות ollama שנלקחו מהסרטון
- הסבר מילולי קצר: "מה החלטה הטכנית שהוידאו חיעע, ומה עשייתך אחרת בתת-ניסוי 2/3/4?"

## **8 סיכום הניסוי**

הניסוי קצר (מסמך קטן, מעט שאילתות) אך עשיר בוריאציות מימוש.

### **1.8 כל תת-ניסוי**

- משתמש באותו pipeline בסיסי (Ollama+LangChain+Chroma+Embeddings)
- משנה רכיב/פרמטר אחד ובודד השפעה טכנית (Latency, שגיאות, RAG failures)

## **2.8 התוצאות**

הסטודנט לא רק "יודע מה זה RAG ו-Ollama-", אלא מבין:

- אין התקנה/קונפיגורציה משفيיעת
- מה האלטרנטיבות המקומיות/ウンנות
- למה המלצות בסרטון/ספר הן כאלה, ומה המחיר של לסתות מהן

## **3.8 הצגת התוצאות**

על הסטודנט לתרגם ולחשוב באיזה אופן מושכנע להציג את תוצאות הניסוי, החקירה של הניסוי, והמסקנות מהניסוי. מומלץ לתקן את התוצאות בגרפים לפי שיקול דעת הסטודנט.

---

הערה: מסמן זה כתוב בלשון זכר מטעמי נוחות בלבד, אך מיועד לנשים ולגברים כאחד.