

LLMs בסביבה מרובת סוכנים – הגשת מטלה 6 – הנדסת פרומפטים

שם קוד קבוצה – 1998

ת"ז ושם – 206585614, טל ברדה

תאריך הגשה – 17.12.25

קישור לריפוזיטורי ב-Github –

<https://github.com/TalBarda8/prompt-optimization-lab>

המלצה לציון עצמי על ההגשה – 97

הצדקה להערכה העצמית -

אני מעניק לעבודה ציון 97/100, משום שהיא עומדת במלואה בכל 147 דרישות המפרט האקדמי, ומפגינה רמה גבוהה של מצינויות בכל היבטי הפרויקט. נקודות החוזק המרכזיות כוללות ארכיטקטורה מודולרית מתוכננת היטב המבוססת על תבנית Building Blocks עם שישה רכיבים ניתנים להרכבה; תיעוד ארכיטקטוני מקיף הכולל שלוש דיאגרמות (C4 (Context, Container, Component) המוטמעות ב- ARCHITECTURE.md באמצעות Mermaid; ומערך בדיקות אוטומטיות רחב היקף הכולל 357 בדיקות עם כיסוי של 72%.

המערכת תומכת בכמה ספקי LLM (OpenAI, Anthropic, Ollama) דרך ממשק אחיד, וכוללת גם יכולת הורדה אוטומטית של מודלי Ollama שאינם קיימים מקומית. בנוסף, היא מחשבת מטריקות מתקדמות מתורת האינפורמציה – כמו Entropy, Perplexity ו- Composite Loss – בהתאם לנוסחאות מתמטיות מדויקות. הפרויקט ממשש שבע טכניקות הנדסת פרומפטים (Baseline, CoT, CoT++, ReAct, ToT, Role-Based, Few-Shot), כולל תמיכה במצב מהיר עבור מודלים מקומיים, וכן מייצר יותר מ-12 ויזואליזציות איכותיות באמצעות Matplotlib ו-Seaborn.

לצד החוזקות, יש גם נקודות לשיפור: כיסוי הבדיקות במספר מודולים, כגון orchestrator_evaluation.py (שנמצא על 19%), יכול להיות גבוה יותר; ניתן היה להוסיף בדיקות אינטגרציה מקיפות מול LLM אמיתיים; וכן לשקול פיתוח מערכת לוגים וניטור מובנית יותר.

ההשקעה בפרויקט הייתה משמעותית, וכללה למעלה מ-40 שעות עבודה שהתפרשו על תכנון ארכיטקטוני מעמיק, יישום איטרטיבי עם רפקטורינג מתמשך (לדוגמה, orchestrator.py ירד מ-450 שורות ל-94 פקודות), כתיבת תיעוד מפורט מאוד (README בן 1,056 שורות, PRD מקיף, מדריכי קונפיגורציה), ותהליך QA קפדני.

האלמנטים הייחודיים בפרויקט כוללים מימוש של תבנית Building Blocks עם חוזים ברורים, תמיכה ב-Fast Mode הכולל אופטימיזציות ייעודיות ל-LLM מקומיים (כמו Timeout קצר יותר והנחיות מקוצרות), וכן שילוב ישיר של דיאגרמות C4 ב-Markdown באמצעות Mermaid להצגה טבעית ב-GitHub.

לאורך הפיתוח למדתי עקרונות חשובים של תכנון ארכיטקטוני מתקדם, חשיבות הפרדה ברורה של אחריות בקוד, שיטות לבדיקות אוטומטיות מקיפות, ומתודולוגיות להערכת מודלי שפה גדולים באמצעות תורת האינפורמציה. בנוסף, התחזקתי בהבנה של חשיבות תיעוד איכותי ויצירת ויזואליזציות ברורות ומדויקות לתוצאות מחקר אקדמי.

