

SYSTEMS BASED ON MULTIPLE SOURCES – PAPER 6 – THE PRESENTATION

שם קוד קבוצה – 1998

ת"ז ושם – 206585614, טל ברדה

תאריך הSENT – 17.12.25

קישור לריפורזיטורי ב-Github

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

המלצה לציון עצמי על הSENT – 97

הצדקה להערכת העצמיה –

אני מעניק לעבודה ציון 97/100, משומש שהוא עומד במלואה בכל 147 דרישות המפורט האקדמי, ומפגינה רמה גבוהה של מציאות בכל היבטי הפרויקט.

נקודות החזק המרכזיות כוללות ארכיטקטורה מודולרית מתוכנתית היבט המבוססת על תבניות Building Blocks עם שישה רכיבים ניתנים להרכבה; תיעוד ארכיטקטוני מקיים הכלל שלוש דיאגרמות C4 (Context, Container, Component ARCHITECTURE.md; Mermaid באמצעות בדיקות Mermaid; ומערך בדיקות אוטומטיות רחב היקף הכוללת 357 בדיקות עם CISCO של 72%.

המערכת תומכת בכמה ספקים (OpenAI, Anthropic, Ollama) דרך ממשק אחד, וככלות גם יכולת הורדה אוטומטית של מודלי Ollama שאינם קיימים מקומיות. בנוסף, היא מחשבת מטריקות מתකומות מתורת האינפורמציה — כמו Entropy, Perplexity ו- Composite Loss — בהתאם לנוסחאות מתמטיות מדויקות. הפרויקט ממש שבע טכניקות הנדסת פרומפטים Baseline, CoT, CoT++, ReAct, ToT, Role-Based, Few-Shot, כולל תמייה במצב מהיר עבור מודלים מקומיים, וכן מייצר יותר מ-12 ויזואלייזציות.

איך אינטגרציה Matplotlib ו-Seaborn.

לצד החזקות, יש גם נקודות לשיפור: CISCO הבדיקות במספר מודולים, כגון orchestrator_evaluation.py (שנמצא על 19%), יכול להיות גבוהה יותר; ניתן להויסיף בדיקות אינטגרציה מקיפות מול SYSTEMS AMITIES; וכן לשקו פיתוח מערכת לוגים וניתוח מובנית יותר.

ה השקעה בפרויקט הייתה משמעותית, וכללה למעלה מ-40 שעות עבודה שהתרפשו על תכנון ארכיטקטוני עמוק, יישום איטרטיבי עם רפלקטוריון מתמשך (לדוגמה, orchestrator.py ירד מ-450 שורות ל-94 פקודות), כתיבת תיעוד מפורט מאוד README בין 1,056 שורות, PRD מקיים, מדריכי קונפיגורציה, ותהליך QA קפדי.

האלמנטים הייחודיים בפרויקט כוללים שימוש בתבניות Building Blocks עם חזקים בורות, תמייה Fast Mode היכול אופטימיזציות ייעודיות ל- SYSTEMS מקומיים (כמו Timeout קצר יותר והנחיות מקוצרות), וכן שילוב ישיר של דיאגרמות C4 ב- Markdown באמצעות Mermaid GitHub להציג טבויות.

לאורך הפיתוח למדתי עקרונות חשובים של תכנון ארכיטקטוני מתקדם, חשיבות הפרדה ברורה של אחראיות בקוד, שיטות לבדיקות אוטומטיות מקיפות, ומתודולוגיות להערכת מודלי שפה גדולים באמצעות תורת האינפורמציה. בנוסף, התהווות היבט בהבנה של חשיבות תיעוד אינטגרית ויזואלייזציות ברורות ומדויקות לתוצאות מחקר אקדמי.

