

סיכום שיעור

הנדסת פרומפטים למערכות AI בקנה מידה גדול

Chain of Thought, ReAct, Tree of Thoughts

מרצה: ד"ר יoram sagiv

שיעור מס' 6

שיעור זה מושם על הספר של ד"ר יoram sagiv:

**הנדסת פרומפטים: מתמטיקה, שיטה וביצוע
למערכות AI מתקדמות**

Prompt Engineering: Mathematics, Method, and Execution
for Advanced AI Systems

מילות מפתח

- Prompt Engineering -
- Chain of Thought (CoT) -
- ReAct -
- Tree of Thoughts (ToT) -
- Entropy -
- Few-Shot Learning -
- Role-Based Prompting -
- Atomic Prompts -
- Mass Production -
- AI Agents -
- סוכני בינה מלאכותית -

1 מבוא: האתגר של סוכני AI בסקיל

שיעור פותח בדיאו על המצב הנוכחי של תעשיית הבינה המלאכותית. קיימת פער משמעותי בין הقيילות לבין המCEEDות בשימוש בסוכני AI. מצד אחד, ישנה הת啧ומות אקספוננציאלית ביכולות המודלים, אך מצד שני, רק כ-3% מהאפשרויות המלאכותיות ניתנות לאוטומציה מוחלטת.

1.1 הבעה המרכזית

כאשר כותבים פרומפט אחד בבית והוא עובד – זה לא מבטיח הצלחה בסקיל. לצורך קביעת לערתו פניות של ל��חות, הסטטיסטיקה נכנסת לתמונה וכל הפרומפטים עלולים "לקרס". זהה אחת הביעות המרכזיות של חברות פיתוח שמנסות להטמייע AI.

תובנה מרכזית

אנחנו לא מדברים על כתיבת פרומפט לצ'אט פרט. אנחנו עוסקים בהנדסת פרומפטים (Prompt Engineering) עבור סוכנים שצרכים להפעיל את הפרומפט מיליון פעמים ביום. ההבדל הוא קריטי.

2 אנטרופיה – המדריך המרכזי

2.1 מהי אנטרופיה?

אנטרופיה היא מדריך לאי-ודאות או "מבהכה" במערכת. הנוסחה כוללת לוגריתם בסיס 2, המודד את כמות הביטים הנדרשת לייצוג המידע.

- **אנטרופיה גבוהה** – הרבה אפשרויות, המודל "מבולבל"
- **אנטרופיה נמוכה** – מעט אפשרויות, תשובה ברורה וחד-משמעות

2.2 המשמעות לפרומפטים

שշוכבתים פרומפט, השאיפה היא להשיג **אנטרופיה נמוכה**. אם שואלים "כמה זה $2+2$?" רוצים שהמודל יהיה בטוח בתשובה. פרומפט טוב מוביל לאנטרופיה נמוכה, בעוד פרומפט גרווע מוביל לאנטרופיה גבוהה.

כלל מנהה

ביצור בקנה מידה גדול (Mass Production), יש לייצר פרומפטים רבים ולודא שהאנטרופיה נמוכה ככל האפשר – ככל מר, הסוכן מתעקש על אותה תשובה באופן עקבי.

3 פרומפטים אוטומטיים

3.1 הגדרה

פרומפט אוטומי הוא ההוראה הקצרה ביותר שעדיין מבצעת את המשימה המוגדרת. המטרה: מינימום טקסט, מקסימום מידע.

3.2 עקרונות

- יש לפתח שפה מותאמת ומדויקת
- להשתמש במונחים מקובלים כמו "Syntax Parsing" במקום תיאורים ארוכים
- קצר מדי וככליל – לא טוב; ארוך מדי – גם לא טוב
- יש למצוא את האורך האופטימלי הספציפי לכל משימה

3.3 כלל שלושת הפרומפטים

לכל שימוש יש לכתוב שלושה פרומפטים באורכים שונים:

1. פרומפט קצר (כ-50 טוקנים)

2. פרומפט בינוני (כ-200 טוקנים)

3. פרומפט ארוך (כ-500 טוקנים)

יש להשווות ביניהם ולבחרו את האפקטיבי ביותר.

4 – שרשרת מחשבות Chain of Thought (CoT) 4

4.1 הרעיון המרכזי

במקום לבקש מהמודל תשובה ישירה, מבקשים ממנו "לחשב צעד אחרי צעד". זו טכניקה שמשפרת משמעותית את הדיק, במיוחד בעיות לוגיות ומתמטיות.

4.2 דוגמה

שאלה: לדני חמישה תפוחים, הוא זרק שניים ומוצא שלושה. כמה יש לו?
לא CoT: התשובה היא 6.
עם CoT:

1. בהתחלה היו 5 תפוחים

2. הוא זרק 2, אז נשארו 3

3. הוא מוצא 3, אז עכשו יש $3 + 3 = 6$

4. תשובה סופית: 6

4.3 תוצאות מחקריות

החוקרים ב-Google חראו שעל מבחר GSM8K של בעיות מתמטיות, הדיק עליה מ-18% ל-58% בזכות שימוש ב-Chain of Thought.

4.4 הגרסה המשופרת CoT++

הרצת שלושה נתבי חשיבה במקביל וביצוע הצבעת רוב (Majority Voting) לקבלת התשובה הסופית.

4.5 יתרונות וחסרונות

- יתרונות: אפשרויות לדיבוג, השפעה על שלבי ביןיהם, מיקוד המודל

- חסרונות: יותר טוקנים (עלות), יותר זמן תגובה

5 – ReAct – שילוב חשיבה ופעולה

5.1 הרעיון

משלב הסכמה (Reasoning) עם ביצוע פעולות (Acting). המודל לא רק חושב, אלא גם פועל בעולם – משתמש במגוון חיפוש, מחשבונים, או מאגרי מידע.

5.2 מחזור הפעולה

1. **מחשבה (Think)**: מה אני צריך לדעת?
2. **פעולה (Act)**: שימוש בכלים חיצוניים
3. **תצפית (Observe)**: קריית התוצאה
4. חוזרת לשלב 1 עד להשלמת המשימה

5.3 דוגמה מעשית

שאלה: האם יורד גשם בעיר הבירה של צרפת?

1. **מחשבה**: אני צריך לדעת מה הבירה של צרפת ואז לבדוק מזג האוויר
2. **פעולה: חיפוש**: "Capital of France"
3. **תצפית: פרייז**
4. **פעולה: חיפוש**: "Paris weather now"
5. **תצפית: שימוש**, 22 מעלות
6. **תשובה סופית**: לא, בפריז קרGNU משמש

ReAct 2.0 5.4

גרסה מתקדמת שמוסיפה שלב רביעי – **רפלקציה (Reflect)**. שכבה מטא-קוגניטיבית שבה המודל שואל: האם הפעולה הייתה יעילה? אם לא – משנהים אסטרטגייה.

5.5 יישום מעשי

- מיושם באמצעות שרשרת סוכנים. כל סוכן עושה שימוש קטנה מאוד:
- סוכן אחד אחראי על גיאוגרפיה
 - סוכן אחר אחראי על מזג אוויר
 - אורקסטראיטור מנהל ונותן את התשובה הסופית

6 – עץ מחשבות – Tree of Thoughts (ToT)

6.1 הרעיון

אם Chain of Thought הוא קו ישיר, המודל חוקר מספר נתבי חשיבה במקביל, כמו שחמיטאי שחשوب על עשרה מלחכים קדימה.

6.2 התהילה

1. **יצירת ענפים:** המודל מציע מספר רענוןנות
2. **הערכתה (Evaluation):** מתן ציון לכל כיוון
3. **בחירה:** המשך בנתיב המבטיח ביותר
4. **גיזום (Pruning):** חיתוך ענפים עם ציון נמוך

6.3 תוצאות מחקריות

על משחק "Game of 24":

– הצלחה של 4% עם GPT-4: Chain of Thought –

– הצלחה של 74% עם GPT-4: Tree of Thoughts –

קפיצה של פי 18!

6.4 גיזום דינמי

עד יכול לגדול ללא גבולות. לכן משתמשים ב:

– הגבלת מספר צמתים – Budget Forcing –

– חיתוך ענפים לא מבטיחים – Dynamic Pruning –

7 פרומפטים מבוססי תפקיד – Role-Based Prompting 7

7.1 הרעיון

למודל אין "אישיות", אבל אפשר לתת לו **תפקיד**. התפקיד משפיע על אופי התשובות.

7.2 דוגמאות

- "אתה פרופסור לכלכלה"
- "אתה מתכנת מומחה"
- "אתה מורה לילדים"

7.3 מתי זה עובד?

התפקיד עוזר כשהוא **ספציפי** ורלוונטי למשימה. "אתה מומחה" פחות טוב מ"אתה מומחה לכלכלה הantinegotiata שלמד באוניברסיטה".

8 יסודות מתמטיים – הרחבה

סעיף זה מסכם את הבסיס המתמטי של הנדסת פרומפטים, המבוסס על תורה האינפורמציה.

8.1 פונקציית הפסד לפרוומפט

פונקציית ההפסד המרכזית משלבת שלושה מרכיבים:

$$\mathcal{L}_{\text{prompt}} = \alpha \cdot H(Y|x) + \beta \cdot \frac{|x|}{C_{\max}} + \gamma \cdot \text{Perplexity}(x) \quad (1)$$

$H(Y|x)$ – אנטרופיה מותנית (אי-ודאות התשובה בהינתן הפרוומפט)

$\frac{|x|}{C_{\max}}$ – עלות אורך הפרוומפט ביחס למקסימום

$\text{Perplexity}(x)$ – ממד מורכבות/בלבול הפרוומפט

α, β, γ – משקלות לאיזון בין המרכיבים

8.2 אנטרופיה מותנית

האנטרופיה המותנית מודדת את אי-הוודאות בפלט בהינתן הקלט:

$$H(Y|x) = - \sum_{i=1}^n P(y_i|x) \cdot \log_2(P(y_i|x)) \quad (2)$$

המטרה: למזער את $H(Y|x)$ – ככל שהאנטרופיה נמוכה יותר, כך התשובה יותר וודאית ועקבית.

8.3 פרפלקסיטי

מודדות עד כמה המודל "מופתע" מהტקסט: Perplexity

$$\text{PP}(x) = \sqrt[N]{\frac{1}{\prod_{i=1}^N P(w_i)}} \quad (3)$$

פרשנות: פרפלקסיטי נמוכה מעידה שהפרוומפט "אורם" טוב עבור המודל.

8.4 צוואר בקבוק אינפורמטיבי

עיקרונו ה-Information Bottleneck מażon בין דחיסה לשימור מידע:

$$\min_{p(z|x)} I(X;Z) - \beta \cdot I(Z;Y) \quad (4)$$

$I(X;Z)$ – המידע החדי בין הקלט לייצוג (יש למזער – דחיסה)

$I(Z;Y)$ – המידע החדי בין הייצוג לפלאט (יש למакс – שימור)

β – פרמטר איזון בין דחיסה לשימור

ישום: עיקרונו זה מסביר למה פרומוומטים אוטומטיים ממוקדים עובדים טוב יותר – הם שומרים על המידע הרלוונטי תוך הסרת רעש.

9 טיפים מעשיים: Skills ו-Claude-Commands

Skills 9.1

ב-Claude יש מגנון של Skills מוגדרים. מומלץ לתת ל-Claude לכתוב את ה-Skill עבורכם, כי הוא יודע לבנות את ה-Header באופן שמאפשר הפעלה אוטומטית לפי הקשר.

Commands 9.2

במקומות לכתוב את אותו פרומפט כל פעם, אפשר ליצור Commands. הם תומכים בארגומנטים דינמיים וחווסcis זמן.

מטלה לסטודנט

משמעותו של תיאור התובנות האישיות: התיאור הבא הוא **תיאור כללי בלבד**. אנו מוצפים מהסטודנט להביא את התובנות האישיות שלו, את **הפרספקטיבת הייחודית** שלו, ואת **היצירתיות** שלו בהבנת הנושאים המתוארים במטלה.

המטרה היא לא לעקוב אחרי הוראות באופן מכני, אלא להפgin הבנה عمוקה ויכולת יישום עצמאית.

10.1 מטרת המטלה

להוכיח מה עוזר ומה מקלקל בפרומפט, ולהראות שיפור (או הרעה מכוונת) בביטויים באמצעות גוף.

10.2 שלבי העבודה**10.2.1 שלב 1: ייצור מאגר נתונים**

יש ליצור Dataset של זוגות שאלות-תשובות. דוגמאות:

- **ניתוח סנטימנט**: טקסטים מותזגים כחיובי/שלילי/שמח/עצב
- **תרגילים מתמטיים**: חישובים עם כמה שלבים
- **משפטים לוגיים**: "אם X אז Y, ואם Y אז Z"
- **טייפ לחיסכון בטוקנים**: השתמשו במספטים קצרים.

10.2.2 שלב 2: מדידת בייסליין

הריצו את הנתונים עם פרומפט בסיסי ומדדו:

- מרחקים וקטוריים בין התשובות לתשובות האמת
- היסטוגרמה של המרחקים
- ממוצע ושונות

10.2.3 שלב 3: שיפור הפרומפט

נסו את השיטות הבאות:

1. **שיפור פרומפט רגיל** – שינויים בניסוח הסיסטם

2. **Few-Shot Learning** – הוספת דוגמאות (עד 3 דוגמאות)

3. **"חשוב צעד אחריו צעד"** – **Chain of Thought**

4. **(אופציונלי) ReAct** – שילוב עם כלים חיצוניים

10.2.4 שלב 4: השוואת הציגות

יש להציג גרפ שומרה את השיפור (או ההרעה) בין הגרסאות השונות של הפרומפט.

10.3 מה אנחנו מצלפים לראות

ציפיות מהסטודנט

- **יצירתיות:** בחירה מקורית של תחום או בעיה לבדיקה
- **תובנות אישיות:** הסברים **משלכם** למה שינוי מסוים שיפור או הרע
- **פרספקטיבית יהודית:** גישה אישית לפתרון הבעיה
- **ניסויים עצמאיים:** מעבר להוראות הבסיסיות – חקרו!
- **חשיבה ביקורתית:** האם המתודולוגיה עבדה? למה? למה לא?

10.4 הערות חשובות

- לא חובה, אך מומלץ מאוד למינידה **ReAct**
- השתמשו בספר ההרצאה כבסיס לחיפוש וكمוקור מילוט מפתח
- הכמות היא לטובתכם, אך התחשבו בمبرallocות טוקנים
- **זכרו:** אנחנו מדברים על **Mass Production**, לא על שימוש בודד

11 סיכום

1. **הנדסת פרומפטים (Prompt Engineering)** היא לא כתיבת שאלות לצ'אט – זו הנדסה לייצור בסקיל
2. **אנטרופיה נמוכה** היא המטרה – תשובה עקבית וברורה
3. **Chain of Thought** משפר דיקוק בבעיות לוגיות ומתמטיות
4. **ReAct** מאפשר למודל לפעול בעולם האמיתי
5. **Tree of Thoughts** מאפשר חקירה של מספר נתיבים במקביל
6. **סוכנים קטנים ורזים** בשרשרת – טובים יותר מסוכן אחד גדול
7. **מדידה סטטיסטית** היא הכרחית לבדיקת איכות הפרומפטים