

סיכום שיעור

הנדסת פרומפטים למערכות AI בקנה מידה גדול

Chain of Thought, ReAct, Tree of Thoughts

מרצה: ד"ר יורם סגל

שיעור מס' 6

שיעור זה מבוסס על הספר של ד"ר יורם סגל:

הנדסת פרומפטים: מתמטיקה, שיטה וביצוע

למערכות AI מתקדמות

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

מילות מפתח

- Prompt Engineering - הנדסת פרומפטים
- Chain of Thought (CoT) - שרשרת מחשבות
- ReAct - שילוב חשיבה ופעולה
- Tree of Thoughts (ToT) - עץ מחשבות
- Entropy - אנטרופיה, מדד לאי-ודאות
- Few-Shot Learning - למידה מדוגמאות מעטות
- Role-Based Prompting - פרומפטים מבוססי תפקיד
- Atomic Prompts - פרומפטים אטומיים
- Mass Production - ייצור בקנה מידה גדול
- AI Agents - סוכני בינה מלאכותית

1 מבוא: האתגר של סוכני AI בסקייל

השיעור פותח בדיון על המצב הנוכחי של תעשיית הבינה המלאכותית. קיימת פער משמעותי בין הציפיות לבין המציאות בשימוש בסוכני AI. מצד אחד, ישנה התקדמות אקספוננציאלית ביכולות המודלים, אך מצד שני, רק כ-3% מהמשימות המלאות ניתנות לאוטומציה מוחלטת.

1.1 הבעיה המרכזית

כאשר כותבים פרומפט אחד בבית והוא עובד – זה לא מבטיח הצלחה בסקייל. כשצריך לענות למיליון פניות של לקוחות, הסטטיסטיקה נכנסת לתמונה וכל הפרומפטים עלולים "לקרוס". זוהי אחת הבעיות המרכזיות של חברות פיתוח שמנסות להטמיע AI.

תובנה מרכזית

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

2 אנטרופיה – המדד המרכזי

2.1 מהי אנטרופיה?

אנטרופיה היא מדד לאי-ודאות או "מבוכה" במערכת. הנוסחה כוללת לוגריתם בבסיס 2, המודד את כמות הביטים הנדרשת לייצוג המידע.

– **אנטרופיה גבוהה** – הרבה אפשרויות, המודל "מבולבל"

– **אנטרופיה נמוכה** – מעט אפשרויות, תשובה ברורה וחד-משמעית

2.2 המשמעות לפרומפטים

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

כלל מנחה

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

3 פרומפטים אטומיים

3.1 הגדרה

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

3.2 עקרונות

– יש לפתח שפה מתומצתת ומדויקת

– להשתמש במונחים מקובלים כמו "Syntax Parsing" במקום תיאורים ארוכים

– קצר מדי וכללי – לא טוב; ארוך מדי – גם לא טוב

– יש למצוא את האורך האופטימלי הספציפי לכל משימה

3.3 כלל שלושת הפרומפטים

לכל משימה 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 להשוות ביניהם ולבחור את האפקטיבי ביותר.

4 Chain of Thought (CoT) – שרשרת מחשבות

4.1 הרעיון המרכזי

במקום לבקש מהמודל תשובה ישירה, מבקשים ממנו "לחשוב צעד אחר צעד". זו טכניקה שמשפרת משמעותית את הדיוק, במיוחד בבעיות לוגיות ומתמטיות.

4.2 דוגמה

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

1. בהתחלה היו 5 תפוחים
2. הוא זרק 2, אז נשארו 3
3. הוא מצא 3, אז עכשיו יש $3 + 3 = 6$
4. תשובה סופית: 6

4.3 תוצאות מחקריות

החוקרים ב-Google הראו שעל מבחן GSM8K של בעיות מתמטיות, הדיוק עלה מ-18% ל-58% בזכות שימוש ב-Chain of Thought.

4.4 CoT++ – הגרסה המשופרת

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

4.5 יתרונות וחסרונות

- **יתרונות:** אפשרות לדיבוג, השפעה על שלבי ביניים, מיקוד המודל
- **חסרונות:** יותר טוקנים (עלות), יותר זמן תגובה

5 ReAct – שילוב חשיבה ופעולה

5.1 הרעיון

ReAct משלב הסקה (Reasoning) עם ביצוע פעולות (Acting). המודל לא רק חושב, אלא גם פועל בעולם – משתמש במנועי חיפוש, מחשבוניס, או מאגרי מידע.

5.2 מחזור הפעולה

1. **מחשבה (Think):** מה אני צריך לדעת?
2. **פעולה (Act):** שימוש בכלי חיצוני
3. **תצפית (Observe):** קריאת התוצאה
4. חזרה 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 אוויר
2. **פעולה:** חיפוש "Capital of France"
3. **תצפית:** פריז
4. **פעולה:** חיפוש "Paris weather now"
5. **תצפית:** שמש, 22 מעלות
6. **תשובה סופית:** לא, בפריז כרגע שמש

5.4 ReAct 2.0

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

5.5 יישום מעשי

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

6 Tree of Thoughts (ToT) – עץ מחשבות

6.1 הרעיון

אם Chain of Thought הוא קו ישר, Tree of Thoughts הוא מבוך. המודל חוקר מספר נתיבי חשיבה במקביל, כמו שחמטאי שחושב על עשרה מהלכים קדימה.

6.2 התהליך

1. **יצירת ענפים:** המודל מציע מספר רעיונות
2. **הערכה (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":

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

קפיצה של פי 18!

6.4 גיזום דינמי

- עץ יכול לגדול ללא גבולות. לכן משתמשים ב:
- Budget Forcing – הגבלת מספר צמתים
- Dynamic Pruning – חיתוך ענפים לא מבטיחים

7 Role-Based Prompting – פרומפטים מבוססי תפקיד

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 התנהגותית שמלמד באוניברסיטה".

8 יסודות מתמטיים – הרחבה

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

8.1 פונקציית הפסד לפרומפט

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-וודאות התשובה בהינתן הפרומפט)

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

- $\text{Perplexity}(x)$ – מדד מורכבות/בלבול הפרומפט

- 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

Perplexity מודדת עד כמה המודל "מופתע" מהטקסט:

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

פרשנות: פרפלקסיטי נמוכה מעידה שהפרומפט "זורם" טוב עבור המודל.

8.4 צוואר בקבוק אינפורמטיבי

עיקרון ה-Information Bottleneck מאזן בין דחיסה לשימור מידע:

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

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

- $I(Z; Y)$ – המידע ההדדי בין הייצוג לפלט (יש למקסם – שימור)

- β – פרמטר איזון בין דחיסה לשימור

יישום: עיקרון זה מסביר למה פרומפטים אטומיים ממוקדים עובדים טוב יותר – הם שומרים על המידע הרלוונטי תוך הסרת רעש.

9 טיפים מעשיים: Skills ו-Commands ב-Claude

9.1 Skills

ב-Claude יש מנגנון 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.

9.2 Commands

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

מטלה לסטודנט

שימו לב: התיאור הבא הוא תיאור כללי בלבד. אנו מצפים מהסטודנט להביא את התובנות האישיות שלו, את הפרספקטיבה הייחודית שלו, ואת היצירות שלו בהבנת הנושאים המתוארים במטלה.

המטרה היא לא לעקוב אחרי הוראות באופן מכני, אלא להפגין הבנה עמוקה ויכולת יישום עצמאית.

10.1 מטרת המטלה

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

10.2 שלבי העבודה

10.2.1 שלב 1: יצירת מאגר נתונים

יש ליצור Dataset של זוגות שאלות-תשובות. דוגמאות:

- ניתוח סנטימנט: טקסטים מתויגים כחיובי/שלילי/שמח/עצוב
- תרגילים מתמטיים: חישובים עם כמה שלבים
- משפטים לוגיים: "אם X אז Y, ואם Y אז Z"
- טיפ לחיסכון בטוקנים: השתמשו במשפטים 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. Chain of Thought - "חשוב צעד אחרי צעד"
4. ReAct (אופציונלי) - שילוב עם כלים חיצוניים

10.2.4 שלב 4: השוואה והצגה

יש להציג גרף שמראה את השיפור (או ההרעה) בין הגרסאות השונות של הפרומפט.

10.3 מה אנחנו מצפים לראות

ציפיות מהסטודנט

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

10.4 הערות חשובות

- ReAct - לא חובה, אך מומלץ מאוד ללמידה
- השתמשו בספר ההרצאה כבסיס לחיפוש וכמקור מילות מפתח
- הכמות היא לטובתכם, אך התחשבו במגבלות 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 בסקייל
2. **אנטרופיה נמוכה** היא המטרה - תשובות עקביות וברורות
3. Chain of Thought משפר דיוק בבעיות לוגיות ומתמטיות
4. ReAct מאפשר למודל לפעול בעולם האמיתי
5. Tree of Thoughts מאפשר חקירה של מספר נתיבים במקביל
6. **סוכנים קטנים וריזים** בשרשרת - טובים יותר מסוכן אחד גדול
7. **מדידה סטטיסטית** היא הכרחית לבדיקת איכות הפרומפטים