התפתחות מנגנוני תיאום כלים מרובים בסוכני LLM

אבי איילי — ת.ז. 300228160
קורס: אורקסטרציה של סוכני AI
מרצה: ד"ר יורם סגל
2026

מסמך זה נוצר בסיוע בינה מלאכותית

תוכן העניינים

1 יסודות סוכני שפה גדולים 3
1.1 הצורך בסוכנים המסוגלים להשתמש בכלים 3
1.2 ארכיטקטורה בסיסית של סוכנים 3

2 ארכיטקטורה 5
2.1 הקדמה אדריכלית 5
2.2 עקרונות ערוצים מובנים 5
2.3 הפרדה היררכית ותיאום רובוטי 5
2.4 עיצוב ממשקים טרנספורמטיביים 6
2.5 יעילות תכנוני קדימות 6
2.6 סיכום ומשמעויות 7

3 תזמור כלים מרובים: מנגנונים וזרימות עבודה 8
3.1 בחירת כלים 8
3.2 זרימת ביצוע 8
3.2.1 נוסחה להערכת סדר התזמור 8
3.3 ניהול מצב 9
3.3.1 קוד דוגמה 9

4 יישומים ועדונות 10
4.1 יישומים מעשיים 10
4.2 השוואת מודלים 10
4.3 עדונות 10
4.4 ניהול זיכרון 11
4.5 בחירת כלים מודעת להקשר 11
4.6 וידוא תוצאות כלים 11
4.7 דפוסי בדיקה 11
4.8 כלי הערכה 11
4.9 מדדי עלות 12

5 הערכה 13
5.1 מבוא להערכה 13
5.2 מטריקות ביצוע 13
5.3 מטריקות בטיחות 13
5.4 דפוסי בדיקה 13
5.5 כלי הערכה 14
5.6 מדדי עלות 14
5.7 דוגמה מעשית 14
5.8 טבלות בנצ'מארק 15

5.9 אתגרים 15

6 סיכום וכיוונים עתידיים 16
6.1 הישגים 16
6.2 מודלי איום 16
6.3 בקרות בטיחות 16
6.4 זיכרון ורעלול 17
6.5 כיוונים עתידיים 17
6.6 סיום 17

פרק 1

יסודות סוכני שפה גדולים

1.1 הצורך בסוכנים המסוגלים להשתמש בכלים

מודלי שפה גדולים (sMLL) הם בעצם מנבאי טקסט שרשור, המעובדים טוב על משימות טקסטואליות טהורות כמו תרגום, סיכום, ודיאלוג, אך במציאות הם חסרים יכולת קריטית: גישה למידע עדכני, ביצוע חישובים מדויקים, או אינטראקציה עם מערכות חיצוניות. כאשר משתמש שואל סוכן "מהו שער השחלף של דולר ליורו היום?" או "אתחל את עבודת קובץ ה-setenrebuK שלי", ה-MLL לבדו אינו יכול לענות—הוא רק מצפה טוב או מניבה טקסט מבוסס על נתוני אימון ישנים.

הפתרון הוא לתת ל-MLL access לכלים חיצוניים: APIs, databases, מחשבונים, מנועי חיפוש, ומערכות תוכנה אחרות. כאשר ה-MLL מזהה שעבור משימה נדרשת פעולה בעולם החיצוני, הוא יכול לבחור כלי מתאים, להשתמש בו, לקלוט את התוצאה, ולהמשיך בתהליך ההנמקה שלו. תבנייה זו—MLL בתור מנהל, כלים בתור ביצוע—הפכה להיסוד של סוכנים אוטונומיים מודרניים.

עם זאת, הוספה של כל כלי חדש דורשת בדרך כלל custom integration code. אם רוצים ל-MLL שלנו להשתמש בגוגל כ-IPA שניהול מטלות יחד עם סוכן נתונים בתוך הארגון, נדרשים שתי חתיכות של לוגיקת כלים, שני סכימות JSON, ושתי קרא-השתלמויות לרעיונות אודות הפורמט של כל כלי וההתנהגות הצפויה שלו. כאשר ניסיון לעבוד עם עשרות כלים, המורכבות הזו הופכת לפורמאלית, ותחזוקה הופכת לחלום של הנדסה.

1.2 ארכיטקטורה בסיסית של סוכנים

ארכיטקטורה אופיינית של סוכן MLL עם תמיכה בכלים מורכבת מחמש רכיבים קריטיים:

1. **לולאת סוכן (pooL tnegA)** — חזרה על מחזור: קרא את המשימה, בחר ביעד, הנמק את הצעדים הבאים, בצע פעולה (או קרא לכלי), טבול את התוצאה, עדכן את הסטאטוס, חזור עד סיום.

2. **בחירת כלים (noitceleS looT)** — כאשר ה-MLL קובע שנדרשת פעולה חיצונית, הוא בוחר כלי מרשימת הכלים הזמינים. בדרך כלל זה שקוף: הכלים מתוארים בשפה טבעית או בסכימה (לדוגמה, OpenAI's function calling schema), והמודל נשמר בהנמקה לגבי כלי הטוב ביותר.

3. **זיכרון וסטאט (etatS dna yromeM)** — סוכנים צריכים לזכור מה שהם כבר בדקו, איזו תוצאות קבלו, והיכן הם בתהליך. זיכרון שטוח (כל התיעוד בהנמקה אחד) נוטה להיכשל בעל משימות ארוכות; מערכות משוכללות משמרות state vectors, knowledge graphs, או מעקבות של כלים קודמים.

4. **טיפול בשגיאות (gnildnaH rorrE)** — כלים עשויים להחזיר שגיאות, ערכים ריקים, או פורמטים בלתי צפויים. סוכנים שרופים צריכים לזהות בעיות, להחזיר ניסיון, או להפעיל קישוטים.

5. **אימות וחמים (sliardrauG dna noitadilaV)** — לא כל החלטה של הסוכן היא בטוחה או חוקית. מערכות בעולם אמיתי דורשות טיפול בהרשאות, אישור אדם לביצוע כזה, ויומן פעילות לגורמים שלטוניים.

הפשוטה ביותר סוכן היא לולאת יחידה עם כלי בודד. פחות פשוט: סוכן עם שלושה או ארבעה כלים, כאשר הבחירה כמו נענוח הדק. והמורכב ביותר: סוכנים מרובים שמתקשרים זה עם זה, כל עם קבוצות כלים שלהם, מתואמים על ידי orchestrator מרכזי שמחליט איזה סוכן לקרא לבצע משימה משנית. זה החלק שנבדוק עוד הלאה: [**erutcetihcrAtnegA_4202_ciporhtnA**].

פרק 2

ארכיטקטורה

2.1 הקדמה אדריכלית

בתחום הסוכנים המבוססים על Large Language Models (MLL), העיצוב האדריכלי של מערכות ההתקשרות עם כלים חיצוניים מהווה נושא קריטי המשפיע ישירות על יעילות, בטיחות וסקלביליות של המערכת כולה. בעוד שמודלי השפה הגדולים מציעים יכולות חישוביות עוצמתיות, הצורך בתיאום יעיל בין רכיבים מרובים — סוכנים, כלים וממשקים — דורש מסגרת ארכיטקטורלית שוטפת וממובנת.

מחקר עדכני בנושא ארכיטקטורות סוכנים מרובי-כלים הדגיש כי הפרדה ברורה בין שכבות של אחריות היא קריטית [**erutcetihcrAtnegA_4202_ciporhtnA**]. במקרה זה, הוא מדגיש כי בעיות תיאום מתעוררות כאשר סוכנים מנסים לתאם פעולות בין מספר כלים חיצוניים, ובהעדר מבנה היררכי ברור, הביצועים משתדרגים לרעה במהירות.

2.2 עקרונות ערוצים מובנים

ה-Model Context Protocol (PCM) מייצג פריצה תאורטית בתחום תקינות ממשקי כלים [**locotorPPCM_4202_ciporhtnA**]. בעוד שמודלים קודמים הסתמכו על ממשקים מותאמים-אישית או על פרוטוקולים אד-הוק, PCM מתאר קבוצה מוגדרת בצורה קפדנית של ערוצי קשר, כל אחד עם סכמה ברורה, טיפוסי נתונים מוגדרים וכללי זרימה סטנדרטיים.

סכמה זו מאפשרת לסוכן MLL לתקשר עם כלים שונים ללא צורך בידע מקדים של הפרטים הייחודיים של כל כלי. במקום זאת, פרוטוקול מובנה זה משדרג את העיבוד:

$$\text{Request} \rightarrow \text{Standardized Interface} \rightarrow \text{Tool Execution} \rightarrow \text{Response Schema} \tag{2.1}$$

תוך כדי יצירת מחסום בטיחותי בין הסוכן לבין הכלים, PCM מצמצם את משטח ההתקפה הפוטנציאלי של הזרקות והתקפות עתידות [**ytiruceSnoitcejnltpmorP_4202_otlA_olaP_ciporhtnA**].

2.3 הפרדה היררכית ותיאום רובוטי

בעיות מורכבות של תיאום מרובי-סוכנים דורשות יותר מממשק סטנדרטי בלבד. מערכות ממשיכות שגדלות בהעדר מבנה היררכי ברור סובלות מעלויות תקשורת מוגברות, עיכובים במתן החלטות, ועלייה במספר ההחמרות של משאבים [**LRtnegAitluMlacihcrareiH_3202_TlM_drofnatS**].

מספר ארכיטקטורות אחרונות הציעו דוגמה לתיאום היררכי, בו סוכן "מנהל" מתאם בין סוכנים עובדים מסוגים שונים, וכל עובד דן בקבוצה ספציפית של כלים או משימות. דפוס זה משפר משמעותית את ביצועי המערכת הכוללת.

דוגמה בולטת היא גישת ה-"דלגציה היררכית" המיושמת במסגרת CrewAI [**eleDlacihcrareiH_4202_lAwerC**] בה סוכן מנהל מרכזי משחק תפקיד דומה לא"מתעתס" או "אדריכל משימות" שמפצל בעיות גדולות לתתמשימות קטנות יותר, מקצה כל אחת מהן לסוכן עובד מתמחה, וצובר את התוצאות ממנו.

$$\text{Manager Agent} \xrightarrow{\text{Task Decomposition}} \text{Worker Agent}_1, \text{Worker Agent}_2, \ldots, \text{Worker Agent}_k \tag{2.2}$$

דוגמה מתמטית של מנגנון דלגציה זה:

$$\text{ComplexTask} = \bigcup_{i=1}^{n} \text{SubTask}_i, \quad \text{Quality}_{\text{final}} = \frac{1}{n} \sum_{i=1}^{n} \text{Quality}(\text{SubTask}_i) \tag{2.3}$$

גישה זו מוודאת כי כל סוכן עובד מתמחה בתחומו, וכלל המערכת מתנהלת בצורה סדירה ויעילה יותר.

2.4 עיצוב ממשקים טרנספורמטיביים

התפקיד החיוני של ממשקים בעיצוב מערכות סוכנים הוא לא רק להעביר מידע, אלא לתמרן אותו בדרכים שהופכות אותו לשמיש עבור השלב הבא של המערכת. הדמיון ל"טרנספורמר" (במובן של הפיכה או שינוי) טמון בעצם העובדה שכל ממשק מבצע טרנספורמציה על הנתונים.

בעיזבון OpenAI's Agents SDK, שילוב של MCP מראה כיצד יכול למנוע טרנספורמציות נתונים מסוג זה להיות ממובנות בחוזה [**noitargetnIPCMstnegA_4202_lAnepO**]. כל מצרף כלי מתואר בסכמה, וכל קריאה לכלי, כל הודעת תגובה, וכל שגיאה עוקבת הסכמה זו. זה מנע בעיות ספציפיות של חוסר-התאמה בינו כלים דומים אך לא זהים.

$$\text{Input Schema} \rightarrow \text{Tool Transformation} \rightarrow \text{Output Schema} \tag{2.4}$$

שלב זה של סטנדרטיזציה מצביע על הבנה עמוקה: כל כלי אינו מבודד. הוא קטן ברשת גדולה יותר של שיתופי פעולה שבהם צורת הפלט של כלי אחד עשויה לשמש כקלט לכלי אחר. לכן, הגדרת סכמות קפדנית היא לא רק בעיה של הנדסת תוכנה טכנית — היא קביעה של הנושא האדריכלי הבסיסי של המערכת כולה.

2.5 יעילות תכנוני קדימות

מערכות סוכנים אמיתיות דורשות לא רק ממשקים תקניים אלא גם אסטרטגיות אמיתיות לתכנון משימות ותיאום משאבים. בתחום למידה מחוזקת מרובי-סוכנים, נמצא כי טכניקות תכנון היררכי משפרות משמעותית את ביצועי המערכת [**LRtnegAitluMlacihcrareiH_3202_TIM_drofnatS**].

כאשר סוכן מנהל פשוט מחליט להקצות משימה, הוא אינו בוחר רק איזה עובד יקבל אותה, אלא גם באילו פרמטרים או כלים יש להשתמש. הבחירה הזו יכולה להיות מתוך מגוון גרוע רחב:

$$\text{Utility} = f(\text{Task Complexity}, \text{Worker Skill}, \text{Tool Availability}, \text{Time Constraint}) \tag{2.5}$$

דוגמה מעשית: אם משימה דורשת קריאה לבסיס נתונים ואז עיבוד הנתונים, מנהל תכנון יטוב יבחר בעובד בעל גישה פנימית לכלי בסיס הנתונים, וישתמש בכלי עיבוד נתונים שהוכח כיעיל לסוג הנתונים ההוא.

2.6 סיכום ומשמעויות

ארכיטקטורה חזקה של מערכת סוכנים מתעוררת מתוך שילוב של שלושה אלמנטים: (1) פרוטוקולים מובנים וסטנדרטיים לתקשורת בין סוכנים וכלים, (2) מבנה היררכי ברור הממונע את תיאום ואת מינוף משאבים, ו-(3) תכנון אמיתי של קדימות וקצאה של משימות המבוסס על מדדים כמותיים.

המחקר שלנו מתמקד בחקר בעומק של אלמנטים אלו, תוך שימוש בטכניקות בדיקה השוואתיות לזיהוי עיצובים אופטימליים ודפוסים של כישלון משותפים.

פרק 3

תזמור כלים מרובים: מנגנונים וזרימות עבודה

3.1 בחירת כלים

בחירת אילו כלים להשתמש היא קריטית להצלחת סוכן. מנגנון תשומת הלב של Transformer מאפשר לסוכן לחשוב אילו כלים רלוונטיים בהקשר הנוכחי, דרך מנגנון של attention weights המדגישים את הכלים הרלוונטיים ביותר.

3.2 זרימת ביצוע

זרימת הביצוע של סוכן תזמור כלים כוללת את השלבים הבאים:

1. קבלת קלט ממשתמש
2. קדוד הקלט לרכיבי סמנטיים עם embeddings
3. שימוש במנוע חשיבה (LLM) בחירת כלים הבאים
4. ביצוע הכלים בסדר הנקבע
5. איסוף תוצאות והעדכון של מצב הסוכן
6. חזרה לשלב 3 עד להשלמת המשימה

3.2.1 נוסחה להערכת סדר התזמור

ניתן לתאר את איכות סדר תזמור על ידי:

$$\text{score} = \frac{1}{n}\sum_{i=1}^{n} r_i - \lambda \cdot \text{cost}_i \tag{3.1}$$

כאשר r_i הוא התגמול (reward) מביצוע כלי i, cost_i הוא העלות החישובית של הכלי, ו-λ הוא פרמטר איזון.

3.3 ניהול מצב

ניהול מצב (state management) בסוכנים תזמור כלים חייב לתמוך בתכונות הבאות:

- **עדכון מצב אטומי**: תוצאות כלים חייבות להיות משולבות בצורה עקבית למצב הגלובלי
- **היסטוריית מעקב**: שמירה על רשומה מלאה של כל קריאות הכלים לדוגמה (debugging) ושקיפות
- **מחזוריות**: זיהוי כאשר סוכן קורא לאותו כלי בשוב ושוב ללא התקדמות, וביצוע מנגנוני עצירה

3.3.1 קוד דוגמה

להלן דוגמה של קוד Python המממש בחירת כלים בסיסית:

```
def select_tool(state, available_tools):
    """יחכונה בצמה דמס לע ילכ רחב"""
    embeddings = encode_state(state)
    scores = []
    for tool in available_tools:
        score = similarity(embeddings, tool.embedding)
        scores.append((tool, score))

    best_tool, best_score = max(scores, key=lambda x: x[1])
    return best_tool
```

קוד זה מדגים את ההיגיון של בחירת כלים: אנחנו מחשבים דמיון בין המצב הנוכחי לכל כלי זמין, ואז בוחרים את הכלי בעל הדמיון הגבוה ביותר.

פרק 4

יישומים ועדונות

4.1 יישומים מעשיים

סוכנים בעלי יכולות תזמור כלים מעוצבים משמשים כיום במגוון יישומים:

- **מערכות שאלות ותשובות**: סוכנים המחפשים מסמכים, מחלצים מידע רלוונטי, וסוכמים תשובות
- **אסיסטנטים ניתוח נתונים**: המחזיקים וביצוע שאילתות בסיסי נתונים, פעולות סטטיסטיות, וויזואליזציות
- **תזמורנים של משימות סדורות**: ניהול זרימות עבודה מורכבות עם תלויות בין כלים

4.2 השוואת מודלים

בטבלה 4.1 נראה השוואה של מודלים שונים המשמשים בסוכנים מודרניים:

טבלה 4.1: השוואת מודלים לתזמור כלים

Model	Parameters	Score	Notes
BERT-base	110M	84.7%	Encoder-only
GPT-3	175B	88.2%	Decoder-only
T5-large	770M	86.1%	Encoder-Decoder
LLaMA-2-7B	7B	85.9%	Open source
Claude-2	100B+	90.5%	State of art

4.3 עדונות

עדונות (fine-tuning) בסוכנים מכללות הדרכה על משימות ספציפיות לתחום בעלות בנטיביות נמוכה:

1. בחרו תרחיש משימה ספציפי
2. איסוף סט נתונים של דוגמאות input-output
3. הדרכה של המודל בסוג הפסד מותאם
4. הערכה על סט בדיקה עצמאי

4.4 ניהול זיכרון

זיכרון בסוכנים יכול להיות:

- **זיכרון קצר טווח**: הקשר של השיחה הנוכחית בלבד
- **זיכרון ארוך טווח**: הרחבת זיכרון שיח עם vector databases
- **זיכרון עקרונות**: מידע מובנה על אילוצים, מטרות, או זהות הסוכן

4.5 בחירת כלים מודעת להקשר

בחירת כלים מודעת להקשר משתמשת בעדויות סטטיסטיות כדי להחליט אילו כלים סביר להיות שימושיים. לדוגמה, אם המשתמש שואל שאלה על מניות, סוכן עשוי להתייחס להסתברות גבוהה יותר לכלי stock market API.

4.6 וידוא תוצאות כלים

חשוב לתקף תוצאות מכלים כדי להבטיח שהן הגיוניות:

- **בדיקות סוג**: וודא שסוג פלט הכלי תואם לצפוי
- **בדיקות תוקף**: וודא שערכים נמצאים בטווח הגיוני
- **בדיקות עקביות**: בדוק שתוצאות כלים מרובים שאנו צופה שיהיו עקביות, באמת

4.7 דפוסי בדיקה

דפוסי בדיקה נפוצים עבור סוכנים כוללים:

1. **בדיקה משלית**: בחן תרחישים פשוטים עם תוצאות צפויות ידועות
2. **בדיקה רגרסיה**: אשר שמודל לא היפך על בעיות שלפני הוא פתר
3. **בדיקה במקרה קצה**: בדוק עם קלט קיצוני או לא צפוי
4. **בדיקה קטעים**: מבדוקת של סוכנים חלקיים בנפרד מערכות מלאות

4.8 כלי הערכה

כלים להערכת תפקוד סוכנים כוללים:

- BLEU ו-ROUGE: מטריקות לשיוך טקסט לטקסט
- Accuracy: השחזור של תגובות נכונות על בדיקה סטנדרטית
- F1-Score: ממוצע הרמוני של דיוק ודיוק קבירה
- Latency: זמן לתגובה ממשכנות לזמן תוך כדי

4.9 מדדי עלות

מדדי עלות חשובים לסוכנים:

- **עלות ממוצעת כל בקשה**: סכום אריתמטי של כל קריאות כלים
- **עלות שיא**: הגרוע ביותר הוא מקרה עלות מחשבה חד פעמית
- **עלות מורכבת**: למשל, לקיחת עמלות טוקן ממשלמים API

פרק 5

הערכה

5.1 מבוא להערכה

הערכה של סוכנים תזמור כלים דורשת שיקול קפדני של מטריקות ביצוע, בטיחות, וניתנות לשימוש. בחלק זה נחקור שיטות להערכה שונות וכלים המשמשים בפרקטיקה.

5.2 מטריקות ביצוע

מטריקות הביצוע הראשוניות כוללות:

- **דיוק משימה**: אחוז משימות שהסוכן הושלם בהצלחה
- **מסלול זיכרון**: מספר אינטראקציות הנדרשות בממוצע
- **העלות הקומולטיבית**: עלות חישובית כוללת לסיום משימה
- **זמן תוך כדי**: זמן בעולם בפועל מתחילה להתחזוקה בתרחיש

5.3 מטריקות בטיחות

בטיחות היא חיוני עבור סוכנים בייצור:

1. **עצירה בטוחה**: האם הסוכן עוצר כראוי כשנשאל?
2. **תקיפות למעורבות אדם**: האם הסוכן מחזיק בקשר עם בני אדם כשהוא בתמימות?
3. **קבלת עלות מוגבלת**: האם הסוכן מונע מוציא יתר משום תקציב מסוים?
4. **קביעת מסדרון**: האם הסוכן מעולם עושה דבר שמנוגד למדיניות ארגוני?

5.4 דפוסי בדיקה

דפוסי בדיקה חשובים לתחזוקה אמינה:

- **טסט יחידה**: בדיקה כל סוכן רכיב בנפרד

- **טסט אינטגרציה**: בדיקה סוכנים פעולה עם כלים אמיתיים
- **טסט קצה לקצה**: בדיקה סוכנים על משימות מסתובבות שלמות
- **טסט לחץ**: בדיקה עם קצב גבוה של בקשות כדי לזהות מגבלות

5.5 כלי הערכה

כלים להערכה כוללים:

- MLflow: מעקב אחר ניסויים ותוצאות
- Weights & Biases: ניטור מודל למעקב דרך הדרכה
- Prometheus ו-Grafana: ניטור בזמן אמת של מדדים
- Hugging Face Datasets: בנצ'מארקים סטנדרטיים להערכה

5.6 מדדי עלות

עלות היא גורם קריטי בקביעת כדאיות:

טבלה 5.1: טבלת בנצ'מארק עלות

Approach	Avg Cost	Accuracy	Cost/Acc
ReAct-Base	$0.05	78%	0.064
LangGraph-Pro	$0.12	85%	0.141
Claude-Agent	$0.18	92%	0.196

5.7 דוגמה מעשית

בתרחיש מעשי, כדי להערכת סוכן סוקר מיוחד:

1. אסוף 001 שאלות בדיקה טיפוסיות
2. הפעל סוכן על כל שאלה, רשום קריאות כלים, זמן, וצורך לתוצאה
3. השווה תוצאות לתשובה אדם הערכה
4. חישוב דיוק, זמן ממוצע, עלות ממוצעת
5. חזור על התהליך עם גרסאות מודל שונות

5.8 טבלות בנצ'מארק

טבלאות בנצ'מארק מציגות השוואה של סוכנים מרובים:

טבלה 5.2: בנצ'מארק ביצועים

Agent	Accuracy	Calls	Time
Baseline	65%	150	5.2s
Improved	81%	120	4.8s
State-of-art	89%	95	3.5s

5.9 אתגרים

אתגרים מרכזיים בהערכה כוללים:

- **היעדר בנצ'מארקים טובים**: רבות משימות חדשות אין בנצ'מארקים סטנדרטיים
- **עלות הערכה**: הערכה בנוי סוכנים קרוא לדיינות כדי וקוסטלי
- **השתנות**: תוצאות עשויים לעבור בחילופי קטנים בדר או סדר כלים
- **הערכת בטיחות**: קשה לבדוק בטיחות במלואו בלא בפועל סוכנים

פרק 6

סיכום וכיוונים עתידיים

6.1 הישגים

בתוך מסמך זה סקרנו את התפתחות מנגנוני תיאום כלים בסוכני LLM. הנקודות העיקריות היו:

1. מנגנוני בחירת כלים בולטים בביצועים וגמישות
2. כלים חדשים וגישות קדימה מגבילים זיכרון ועלויות
3. סוכנים קיבלו קוד לשם דוגמה בתחום דרך סטנדרטיזציה

6.2 מודלי איום

בעוד סוכנים תזמור כלים מציעים הרבה עדיפויות, הם כמו כן משלימים איומים:

- **ייצור לא מידי**: סוכנים יכולים להרוצץ פעולות כלים לא מתכוננות
- **פליטת מידע**: עשויים לגלוף מידע רגיש דרך כלים ממושקפים
- **הזרקה**: קלט מטורף יכול לגרום סוכנים לביצוע כלים לא מכוונים

6.3 בקרות בטיחות

בקרות בטיחות צריכות להיות משולבות:

- **רישויים**: בדוק צפיפויות סוכנים בלבד כלים מורשים
- **בדיקות קלט**: טהור ותיקף כל קלט משתמש לפני עירער
- **בדיקות פלט**: בדוק תוצאות כלים לדבר החריג או סכנה
- **גבולות**: הגבל עמלות כלים, זיכרון השימוש, וזמן ביצוע

6.4 זיכרון ורעלול

זיכרון בסוכנים יכול להיות מעוות בשתי דרכים:

1. **ניסיון זיכרון**: גבוה מידי נתונים עלול לגרום לתקלות זיכרון
2. **שקופות עכברים**: סוכנים עשויים לשנות הנתונים תחת הדרך

כדי להפחית סיכונים, נהדר כלים וביצוע זיכרון עם מהקצינויות גבוהים.

6.5 כיוונים עתידיים

טחון המחקר בתיאום סוכנים יעצובו:

- **תיאום אופטימלי**: פיתוח שיטות לניהול סוכנים משימות עם תלויות מורכבות
- **למידה חיזוקים**: שימוש בלמידה חיזוקים כדי לשיפור בחירות כלים
- **סוכנים שתפים**: סוכנים אמת שפעלו אחד לשני כדי לפתור משימות
- **בטיחות מעוררת**: מתודולוגיות חדשות להבטחת בטיחות בסוכנים לתזמור

6.6 סיום

תזמור כלים הוא טוען עולה בתחום עיבוד השפה הטבעית. בעוד שאתגרים רבים נשארו, ההשקעה עדכנית בתחום הביאה לפי ישראל משמעותי.

אנו מצפים שכלים התזמור החדשים ימשיכו להתפתח בעתיד, כשהעמודים מתפתחים עם משימות חדשות וקצפים ודרישות בטיחות. סוכנים שיודעים לתזמר כלים טובים יהיו בעלי ערך רב בתחום בשנים הבאות.

ביבליוגרפיה