



סמינר בנושא מיוחד במדעי המחשב - האוניברסיטה הפתוחה

Large Language Model Agents

מנחה: ד"ר מיה הרמן

מגיש: בוריס ברזנר

תאריך: 25/08/2025



אייר 1 – תמונה שהופקה ע"י מודל גנרטיבי 2 של חברת OpenAI Dall-E עבור הנהניתה:

"Depict an LLM agent running a software company; setting meetings, building software, running finance department, all while making coffee and chaos is in the office"

תוכן עניינים

2	תוכן עניינים.....
3	1. מבוא.....
4	2. חלק תאורטי: המרכיבים של סוכן מודל שפה גדול.....
4	2.1 מודל שפה גדול (LLM).....
4	2.1.1 תקציר ההתפתחות של מודל שפה גדול.....
5	2.1.2 מרכיבי מודל שפה גדול
6	2.1.3 המקודד (Encoder).....
6	2.1.4 המפרש (Decoder).....
6	2.1.5 השנאי (Transformer).....
9	2.1.6 מנגנון הקשב (Attention Mechanism).....
12	2.2 כלים חיצוניים.....
12	2.2.1 מוטיבציה.....
13	2.2.2 שיפור ביצועי מודל בעזרת כלים (ToolFormer).....
16	2.2.3 מפרש קוד (Code Interpreter).....
20	2.3 חשיבה عمוקה.....
20	2.3.1 שרשרת מחשبة (CoT – Chain of Thought).....
21	2.3.2 עץ מחשبة (ToT – Tree of Thoughts).....
22	2.3.3 גרף מחשبة (GoT – Graph of Thought).....
25	2.4 מערכת מרובת סוכנים (Multi-Agent System).....
25	2.4.1 מוטיבציה.....
25	2.4.2 ארכיטקטורות.....
27	3. חלק מעשי: הדגמה – פיתוח מערכת מסחר אלגוריתמי במטבעות קריפטוגרפיים באמצעות מערכת מרובת סוכנים.....
27	3.1 מבוא.....
27	3.2 המערכת AlgoTrader.....
28	3.3 תהליך הפיקוח וגישה שונות למערכות מרובות סוכנים.....
32	4. סיכום ומסקנות.....
33	מקורות.....

1. מבוא

המצאת הגלגל בשנות ה-4000 לפני הספירה שינתה את פני האנושות באופן ממשוני. במקום לסתוכת האנושות על הגב, או על חמור, האדם הרכיב סחרורה כבדה יותר למרכבה והתאמץ פחות במשמעותו. מה היא ההברקה זו שבאזור מוחשכה יצירתיות ומעט הנדסה, חיהם של כלל בני האדם הושפעו, והוא יכול כתפיהם של נושאי המשקל? כמו שהמצאה זו השפיעה על תפוקתו של האדם, גם היום ישנים פיתוחים של טכנולוגיה חדשה המשפיעה בצורה זו – סוכן מודל שפה גדול.

סוכן מודל שפה גדול הינו ישות אוטונומית המבצעת הנקודות מורכבות של משתמש, כאשר התקשרות בין המשתמש לסוכן מתבצעת בשפה טבעית. הסוכן מסוגל לקבל ולהבין הנקודות מורכבות מהמשתמש, לתכנן את אופן הפעולה של עצמו ע"י חלוקת ההנחה לנתן משימות, לבצע את המשימות ע"י שימוש בכלים הקיימים לרשותו, ואם אין לו אותם אז יפתח אותם לעצמו, וכך זאת תוך שהSOCן מודע להקשר בו הוא פועל ומאמת את רלוונטיות הפעולות שלו למשימה. כך, הסוכן הינו עוזר דיגיטלי בעל יכולות גבירותיות בתחוםים רבים ומגוונים, בינהן הנדסת תוכנה, ניתוח נתונים, ניהול יומן פגישות, סיור מוחות, ניהול השקעות פיננסיות ועוד.

בסמינר זה אציג את הטכנולוגיה העומדת בסיס הסוכן, דוגמאות לשימושים עכשוויים, ובוצע הדגמת הרצה של מערכת מרכבת סוכנים שפותחה על ידי קבוצת חוקרים במטרה להחליפן חברת הייטק בקבוצת סוכנים מומחים הפעלים בעצמאות מלאה ובועלות מוגבלות לפטור משימות מורכבות בתחום מדעי הנתונים וכאמור מניבה תוצאות המתחזרות עם פיתוחים בקדמת התחום. פרוייקט הדוגמה - מערכת מסחר אלגוריתמי במטבעות דיגיטליים.

2. חלק תאורי: המרכיבים של סוכן מודל שפה גדול

2.1 מודל שפה גדול (LLM)

סוכן מודל שפה גדול (Large Language Model Agent – LLM Agent) או כפי שנקרא לו "סוכן" בסמינר זה, הינו אוסף של טכנולוגיות החברות זו בזו כזרה חכמה ואלגנטית המפיקות משליהם יכולת פתרון בעיות גבואה, אותה אנו רותמים למען ייעול התפקידה שלנו בעבודה ובחיים האישיים. בין אם אנו מודעים לכך או לא, כמעט בוודאות שמרכיבים מסוימים בחיננומושפעים מחלוקת או כלל הטכנולוגיות שיוזכרו בסמינר זה. השפעה זו באה ידי ביטוי בתחוםים כמו שימוש במונחי חיפוש, ניהול יומן בעזרת פקודות קוליות, בהתקשרות מול תמיית ל��וחות של חברת הטלפון שלנו וועוד.

אבני היסוד של הסוכן:

1. מודל שפה גדול (LLM) – רשת ניירונים גדולה אשר בהינתן קלט של רצף מילים מפיקה את המילה הבאה ברצף בהסתברות גבוהה
2. כלים חיצוניים למודל השפה (Tools) – כל משאב הנitin להפעלה למען קבלת נתונים אשר ישמשו את מודל השפה למtran תשובה אינטואיטיבית יותר לבקשת המשתמש
3. מחשבה عمוקה (Reasoning) – חלוקת תהליכי המענה לשלבים אשר בכל אחד מהם מודא מודל השפה את תקינות וRELATIONSHIPS הפלט להנחה, זאת על ידי ניצול יכולות ההסקה שתמונות בטבע המודול

בסמינר זה נצלול לעומק של כל אחת מאבני היסוד ונפרט את המדע והטכנולוגיה העומדות בכךין. לאורך הסמינר נשלב את אותן אבני יסוד כדי בנייה הדרגתית של קונספט הסוכן. לאחר מכן, נדון במערכת מרחבת סוכנים, ממ"ס (Multi-Agent System, MAS), ולבסוף נבצע הדגמה של ממ"ס ייעודית על מנת לבנות מערכת למסחר אלגוריתמי במתכעות.

2.1.1 תקציר ההתפתחות של מודל שפה גדול

בתחילת המאה ה-20 ניסו מדענים בתחום הפנומנולוגיה (תחום מחקר פילוסופי שניסה להעניק פירוש לוגי מתמטי לתחומי הדעת) באוניברסיטת ג'נבה השוויצרית להביע סמנטיקה של שפה אנושית ע"י חוקים לוגיים. תחום מחקר זה קיבל את התואר "עיבוד שפה טכנית" וכבר הznik תחום שלם של פיתוח ומחקר בנושא זה. (Natural Language Processing – NLP)

בסוף מלחמת העולם השנייה (1945) ניסו הוגי דעתות ומדענים להבין כיצד מחזקים את הסחר הבינלאומי. בעקבות כך, החלה עכודה אקדמית רבה ליצירת מכונה המסוגלת להבין את השפה האנושית, לתרגם אותה לשפת מחשב ובסיום גם לשפות אחרות. אך כפי שהתברר זו אינה ממשה קלה לביצוע. השפה האנושית היא כאוטית (chaotic) ורוב המילים תלויות בהקשרן במיללים אחרים במשפט ואין נושא פירוש חד משמעי וברור.

על מנת השפה האנושית, המתמטיקה נחננת בחוקיות עקבית, משמעות ברורה ובלתי ניתנת לויכוח. בשנות ה-80 פותח במכון מחקר של חברת IBM מודל השפה הסטטיסטי הראשון הפעול באופן איטרטיבי לחיזוי המילה הבא במשפט נתון באמצעות אוצר מילים מוקן מראש. מודל זה אומנם היה מהפכני באותה התקופה, אך ביצועיו נחכמים חלשים מול מודלים מבוססי רשת מקוינטם היום. האתגר המרכזי בשנות ה-80 היה שאין מספיק מידע דיגיטלי הנitinן לשימוש בבניית מודל שפה מורכב יותר.

האינטרנט העולמי (WWW – World Wide Web) הפרק זמין לקהל הרחב בתחילת שנות ה-90 והחל לאגור מאמרם מדעים מכל אוניברסיטאות העולם הנעה תופעה של הצטברות מידע ברשת. זאת, עם הכניסה של יחידות עיבוד גרפיות (GPU – Unit) לשוק (Graphical Processing Unit) ביחיד עם כוח העיבוד הכללי הגובר פי שתים מדי שנתיים (חוק מור) יצרו לראשונה את התנאים ההכרחיים לאימון מודל שפה מורכב המבוסס על רשותות נוירונים עמוקות – מודל השפה הגדול.

2.1.2 מרכיבי מודל שפה גדול

הרכיב העומד ביסודות הסוכן הוא מודל שפה גדול, ולכן העמקה בראשון דורשת דיון שני. טכנולוגיות מודלים לשפה גדולים מקבלות תאוצה פונומנלית בשנים האחרונות, וכל מי שהשתמש בפעם הראשונה בוודאי הופתע מהיכולות של מודלי שפה מפורטים כמו ChatGPT של OpenAI או Claude של Anthropic להבין את ההנחה שלנו כמה כללית, מורכבת, או לא מדויקת שתהיה ולפלוט מענה מדויק.

התפתחויות המובהקות בתחום הבינה המלאכותית התאפשרו בעקבות פריצת דרך בשיטת האימון של מודלי שפה גדולים והתפרנסה במאמר המפורסם "Attention Is All You Need" [1]. במאמר, השתמשו חוקרים מגולב בשיטה ידועה לקודד ולפער מושגים של טקסט אנושי ע"י מנגןן "תשומת לב" או "קשב" (Attention mechanism), תוך האריכתktורה המוצעת במאמר. ארכיטקטורה זו הביאה לשני פריצות דרך משמעותיות:

1. זניחה של פתרונות קודמים כמו רשותות נשנות RNN (Recurrent Neural Networks) או רשותות קונבולוציה, בהם נדרש כוח חישובי משמעותי והכרחיות בחישוב סדרתי

2. מנגןן קשב רב-ראשי (Multi-headed attention mechanism), שהוצע במאמר מאפשר חישוב מקבילי

לארכיטקטורת רשת זו קוראים **השנאי** (Transformer), והוא עומד בסיס ההצלחה של מודלי שפה גדולים, ובפרט בסוכנים. אך לפני שנציגו, נציג שני מרכיבים חשובים המופיעים כמעט בכל ארכיטקטורת מודל שפה (ובפרט בשנאי): **המקודד (Encoder)** וה**מפרש (Decoder)**.

2.1.3 המקודד (Encoder)

הweeney העומד מאחורי המקודד הינו להמיר רצף מילים בשפה אנושית לייצוג וקטורי הכלול בתוכו מידע על שימושות ההקשר בין המילים. לדוגמה, מקודד טוב ידע לייצג את המילה¹ "קציצה" באמצעות וקטורי עבור כל משפט:

1. אני אוהב לאכול קציצה
2. הפסיקתי להתאמן ונהייתי קציצה

במשפט הראשון הכוונה היא لكציצהبشر, ובשני לשון הגה לאדם שהעליה משקל בעקבות היעדר פעילות גופנית.

המקודד בלבד, באמצעות ייצוג וקטורי מקודד שימושות עbor המילים ברצף הקלט, שימושו לביצוע מספר פעולות וביניהן סיווג טקסט, זיהוי סנטימנט, הערכת דמיון בין טקסטים שונים ועוד.

2.1.4 המפרש (Decoder)

באופן משלים למקודד, המפרש ממיר רצף של ייצוגים וקטוריים מקודדי שימושות לשפה אנושית. הוא מבצע זאת באופן איטרטיבי, כך שבכל איטרציה פולט המפרש את המילה בעלת היתכנותה הסתברותית גבוהה יותר מכל המילים במלון² להופיע לאחר פלט הרצף באיטרציה הקודמת. תרגום בין שפות טבעיות, מחולל שפה טבעית, קוד והשלמת טקסט נמנים בתור חילק מהיישומים של רכיב זה.

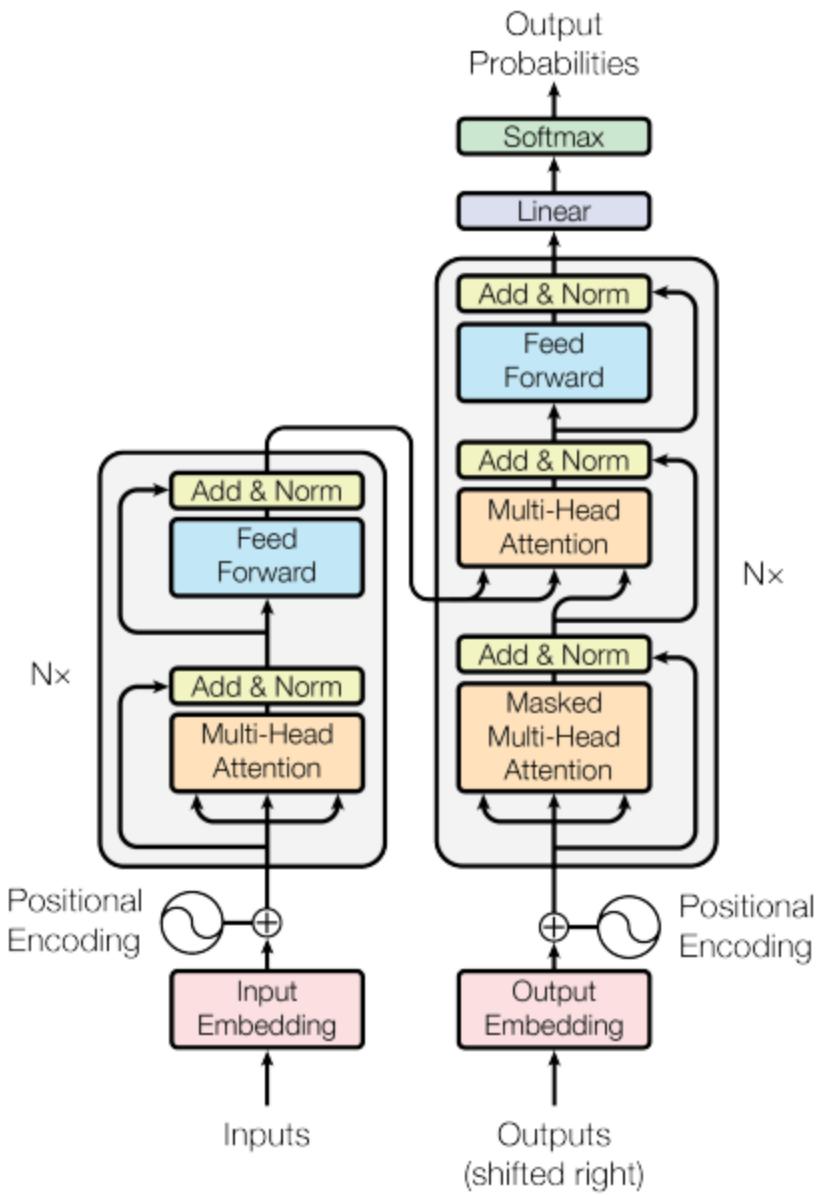
2.1.5 השנאי (Transformer)

השנאי מהווה ארכיטקטורה הכוללת בתוכה מקודד ומפרש. ניתן לחשב על השנאי בתור פונקציה $seq' = seq(t)$ המתקבלת רצף סמלים ($x \dots x_1$) = seq בקלט ופלט רצף סמלים ($y_s \dots y_1$) = $'seq = (y_1, y_2, \dots, y_s)$ לאחר בהתאם לaimon הרשות, לתהיליך זה קוראים התמרת רצף (sequence transduction).

להלן דיאגרמה של ארכיטקטורת השנאי:

¹ בפועל קיים מנגנון הממיר מילים שלמות למיללים חלקיות הנקראים סמלים (tokens), אך בסמינר זה נתיחס למילים שלמות בתור סמלים.

² מלון הינו אוסף כל המילים (סמלים) המופיעים בטבלאת ההטמעה (embedding table) של המודל.



אייר 2 – השנאי כפ' שתואר במאמר [1]

פעולתו של השנאי מתחלקת שני שלבים, שלב הקידוד ושלב הפירוש, ותיאורם מופיע באיור 2 בחלק השמאלי והימני בהתאם. נפרט את פעולתם:

שלב הקידוד

1. הטעמה (embedding): המילים עוברות פונקציה הממירה מילה בשפה אנושית לוקטור d -ממדי הנשלף ממילון שהוכן מראש

$$x'_i = \text{Embed}(x_i)$$

2. קידוד מיקום (positional encoding) – הוקטור \vec{x} מקודד יחד עם מידע על המיקום שלו ברכף

3. המקודד – מחולק לשתי תחת שכבות, האחת מנגנון הקשב והשנייה רשת הזנה قدמית.
באמצעות השניים מפיק המקודד וקטור חדש \vec{z}_i הנושא את משמעות המילה i בהקשר של הרצף בו היא מופיעה (ראה סעיף המקודד)

$$\vec{z}_i = \text{Encoder}(\vec{x}'_i)$$

נדון במנגנון הקשב בהמשך הסעיף.

שלב הפירוש

שלב זה מתחילה עם הזזה ימינה של **פלט המפרש**, פעולה הכרחית להתרעת המפרש.

הזהה ימינה **באייטרציה ראשונה** של המפרש מייצרת סמל תחילת משפט הנקרא <BOS> (Beginning Of Sentence) והוא תוחם צד אחד של הרצף, כאשר אחריו כבר מופיעה המילה הראשונה בחיזוי בעקבות התוצר של המקודד. נציין שבסיום האיטרציה الأخيرة של המפרש נוצרתו משלים הנקרא <EOS>, והוא תומך בעל היכולות הסתברותית.

1. פלט המפרש מוטמע (embedded) ומקודד מיקום בדומה לפעולות הראשונות בשלב הקידוד

2. תת שכבת קשב עם מסכה. מטרתה להתרמי את רצף הפלט של המפרש לכלול מידע על ההקשר של המילים בו. המסכה מונעת מהתהווות שכבה זו לחשב הקשרים של מילים שעתידן להיכנס למפרש – במילים אחרות בכל איטרציה מחושבת טבלאת ההקשרים רק בין המילים בפלט המפרש

3. תת שכבת קשב ללא מסכה (כמו במקודד) המחשבת שוב את רמת הקשב,icut בין המילים בפלט המפרש לבין פלט המקודד

4. שכבת הזנה قدמית לטוויב הנתונים

5. שכבה לינארית

6. פונקציית softmax הידועה שפולטת הסתברויות בין כל הסמלים האפשריים.

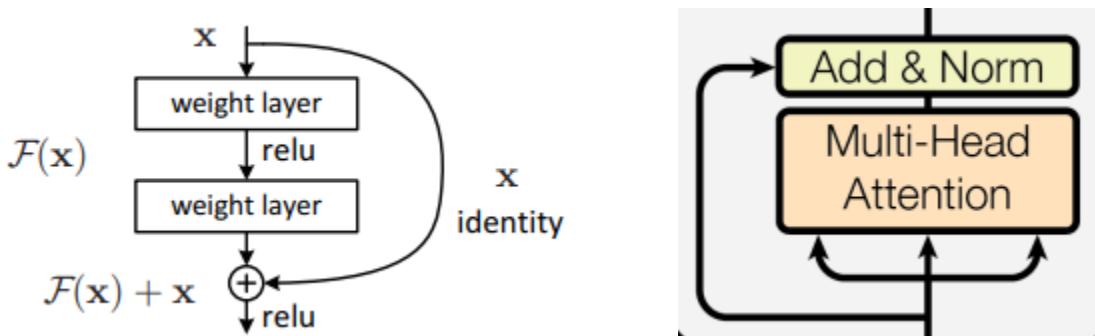
נסכם את פלט שלב הפירוש בכתב מתמטי:

$$y_{i+1} = \text{Decoder}((y_1, \dots, y_i), (z_1, \dots, z_t))$$

כאשר y_i היא מילה ברכף הפלט השני, ו- z_i מילה ברכף הקלט של המפרש.

חיבור שיורי (residual connection)

בארכיטקטורת השנאי ניתן לראות חיבורים מהסוג זהה:



איור 3 חיבור שיורי בשנאי

זהו חיבור שיורי, מטרתו לפתור את בעיית הגרדיאנט הנעלם על ידי סיפוק נתיב ישיר לדילוג הגרדיאנט מעל הפונקציה הנתונה. בנוסף הוא מספק אופטימיזציה לתחילה הלמידה – פונקציית היחידה טריוויאלית למיניהם במקרה זה, נמחיש על ידי הצגת בעיה:

המקרה הלא שיורי – המודול מוצא קירוב טוב לפונקציה $(x)f = y$. נניח כי השגיאה של המודול מינימלית עבור פלט $x = u$, זאת אומרת במקרה שבו f הינה פונקציית היחידה. המודול מאטחל משקלות (פרמטרים) בסביבה קרובה ל-0, ולאחר הרבה הרכבה איטרציות מחפש איזון של המשקלות שיהווה קירוב לפונקציית היחידה – וקרוב לוודאי לא יתכנס אליה.

החיבור השינויי מחפש קירוב לפונקציה $x + (x)f = y$, ומוצא בנקודה את פונקציית המטרה עבור איפוס הפונקציה $0 = (x)f$.

2.1.6 מבנה הקשב (Attention Mechanism)

בלב השנאי, מודל השפה והסוכן, נמצא מבנה הקשב והוא אחראי על הרתיחה המתמטית שמשלב סינטקטיקה, משמעות, והקשר בתור וקטoor, כך שהאינטראקטיה אותו יכולה להרגיש כמו עם בן אדם כאשר בפועל המנגנון הוא אלה מכפלה של מטריצות והרכבה של פונקציה לא לינארית.

תחיליה נציג את היחידה הבסיסית של מבנה הקשב מבוססת מכפלה פנימית ממושקלת (scaled dot product attention).

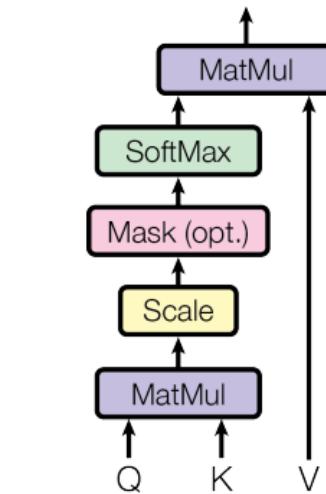
Scaled Dot-Product Attention

מטרת המנגנון היא למצוא מערךות ייחסיטים בתוך הרצף על ידי חישוב "רמת הקשב" (attention score) בין כל זוג מילים ברכף. המטריצות Q , K ו- V (שאילתה, מפתח, וערך) מחושבות באופן הבא:

$$Q = W_Q \cdot E, \quad K = W_K \cdot E, \quad V = W_V \cdot E$$

כאשר E היא מטריצת המילים המוטמעת ברכף הקלט ומטריצות W_x הם הפרמטרים של המנגנון. המנגנון מקיים את המשוואה הכללית הבאה:

$$\text{Attention}(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) \cdot V$$



איור 4 - קשב מכפלה פנימית ממושקלת [1]

משמעות המטריצות:

- Q , שאילתא. עמודה i במטריצה מייצגת את הדרישה של מילה i בחיפוש מילים התומכות במשמעות שלה. ניתן לתאר זאת גם ע"י הצורך שטחית ומייה אחרת גענת אלוי.
- המכפלה הפנימית של שתי העמודות (המילים) תקבע ערך מספרי גבוה.
- K , המפתח. עמודותיה מייצגות את הרלוונטיות שיש למילים להציג. לדוגמה אם העמודה K_i מוקוד משמעות של שם תואר, ועמודה Q_{-i} כלשהי היא שם עצם, תהיה רמת קשב (מידת רלוונטיות) גבוהה בין השניים.
- V , הערך. טומן בתוכו את ההטמעה המקורית של המילה ומיקומה, לאחר התמרה לינארית.

השלבים בחישוב:

1. המכפלה QK^T היא מטריצת דירוג רלוונטיות בין מילים, ערכים גבוהים יותר מסמלים רמת קשב חזקה.
2. המכפלה בסקלר $\frac{1}{\sqrt{d_k}}$ מתמודדת עם בעיית האגדיאנט המתפוצז ונעלם.
3. פונקציית softmax מניבה עבור כל מילה את התפלגות ההסתברויות של מידת הרלוונטיות של מילים אחרות כלפייה.
4. לבסוף, המכפלה במטריצת הערכים V מפרקה רצף סופי בו רמת ההקשר בין כל זוג מילים ברכף משולבים בתוכו.

לסיכום, בתום פעולה מנגנון הקשב מתקבלים אוסף של וקטורים המיצגים את המשמעות ההקשרית ומידתה בין סמל הרצף. זהו ייצוג מתמטי של שפה השואף להתחקות להבנה האנושית.

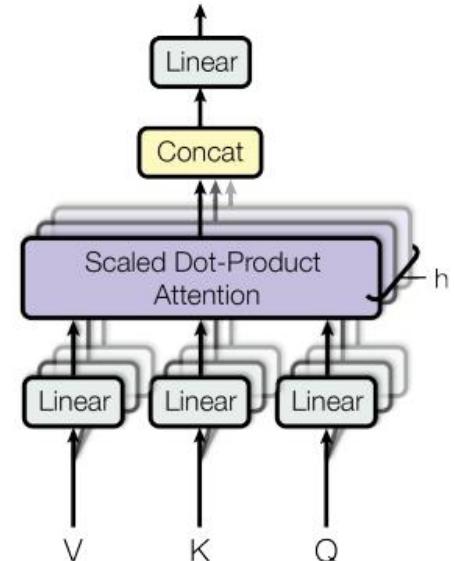
Multi-Head Attention

בחידוש שהוצע במאמר [\[1\]](#) שיכפלו החוקרים את קשב המכפלה הפנימית הממושקלת על מנת לאפשר מקבליות.

בשונה ממנגנון הקשב הבודד שתואר לעיל, שלושת מטריצות הפרמטרים Q , K ו- V מחלקות בגודל לא מטריצות קטנות יותר, כאשר h מייצג את מספר הראשים שיש במנגנון, קרי "מרובה ראשים".

כל ראש של המנגנון מפיק רמות רלוונטיות שונות בין המילים, והפלט של כלל הראשים משורשרים ייחודי לפני שנכנסים לשכבה לינארית אחרונה.

החוקרים הראו שחלוקת זו משווה ביצועים למנגנון הקשב הבודד ואפילו מייצבת את תהליכי הלמידה, כל זאת תוך שמთאפשרת הרצתה מקבילה.



אייר 5 - קשב מרובה ראשיים [\[1\]](#)

לסיכום, מודל שפה גדול הינו כלי לעיבוד שפה טבעית הפולט טקסט בהינתן טקסט. בהינתן מידע טקסטואלי בכמות מסוימת, ניתן לאמן את המודל לרמות גבירות של הבנה, הסקה ותקשורת עד כדי רתימתו לביצוע משימות יומ-יומיות בעבודה ובחיים האישיים (כמו ניסוח מיילים והודעות).

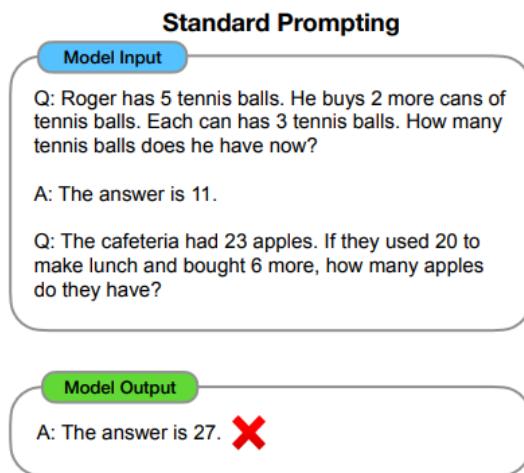
אך בהסתכלות רחבה יותר המודל הוא כשלעצמם אינו אלה מכונות טקסט משוכلات המתחליל ומסיים במיליה. בשאלת על הצעד הבא – האם נוכל לחבר למודל כלים כך שיוכל לבצע בהם שימוש? ובמידה וכן, כיצד? האם ניתן להתמיר מילה בפעולה?

התשובה היא חיובית. במרכיב הכא של הסוקן, **כלים חיצוניים**, נדון בהתמכשות המודל לכלים ונדגים מספר שיטות לכך.

2.2 כלים חיצוניים

2.2.1 מוטיבציה

למטרת היכולות של המודול לייצר טקסט באופן אינטואיטיבי, הוא ניחן בתופעות לוואי לא רצויות. אחת מן התופעות הבולטות במודול היא היזה (hallucination) – תופעה שבה המודול פולט מידע שאינו וaino "מודע" לך. לדוגמה, בדצמבר של שנת 2022 כשןשאל מודול ChatGPT של OpenAI מיהו ראש ממשלה ישראל, היה עונה בckettחה נפתרי בנט, למרות שבנימין נתניהו כיהן בתפקיד באותו עת. דוגמא נוספת להיזה הינה תרגיל מילולי במתמטיקה, ראו איור 6 להלן:



איור 6 הלקוח ממאמר [2] הממחיש שגיאה של המודול בחשיבה אритמטית

שתי הדוגמאות שתוארו לעיל נובעות מסיבות שונות.

- הראשתונה – לא היה ברשותו של המודול את המידע העדכני הנדרש על מנת לענות על השאלה בדבר ראש הממשלה מכחן. המודול אומן על מידע מילולי מהאינטרנט עד שנת 2021, ולכן לא ידע על התחלפות ראשי הממשלה.
- השנייה – המודול לא הצליח לבצע את פעולות האריתמטיקה כנדרש ולכן ענה 27 במקום התשובה הנכונה 9, דבר המעיד על העדר יכולות אריתמטיות.

במילים אחרות המודול אינו מסוגל להתמודד עם כל סוג הבעיות באופן עצמאי וכיים פער ברור בין היכולות של המודול לציפיות שלנו מהביצועים שלו. בדיקן למען גישור פער זה קיימים תחומי מחקרי רחב באקדמיה ובתעשייה העוסק בהعشרת מודול שפה בכליים חיצוניים על מנת לסייע את בעיית ההיזה והמידע החסר.

על מנת לכפר על חולשות המודול נבעצט אינטגרציה בין בין מודלים פשוטים בהרבה מהם למשימות ספציפיות בהן הוא לocketה: מחשבון, גישה למאגר מידע עדכני (כמו ויקיפדיה), גישה לשעון (זמן) ועוד.

כעת נדון בפתרון מעניין ופופולרי הקרי בשם [ToolFormer](#) להתגברות על חולשות המודל בעזרת כלים חיצוניים.

2.2.2 שיפור ביצועי מודל בעזרת כלים – ToolFormer

המאמר [2] מציג את הפרזנטס שבחן היכולות הגוכחות של מודל שפה גדוֹל לפתרון בעיות מורכבות לבין החולשה שלו בפתרון בעיות פשוטות וכפתרון משלב את מודל השפה והמודל פשוט.

המודל המוצע, ToolFormer, משתמש בקריאות API (Application Programming Interface) לשירותים ייעודיים כמו מחשבון, מאגר מידע, לוח שנה (ועוד) במהלך יצירת פלט התשובה ומשלב את התשובות שהתקבלו בפלט. המודל ToolFormer עושה זאת על ידי יצירת סמלים מיוחדים לאזורים בהם דרישה הטערכות של כלי חיצוני, ובהתיקלות בסמלים אלה במצע המודל קרייה לשירות חיצוני בתאם ימישר לחולל את המשך הרצף. הסמלים המיוחדים הם [,] – ➔ – המציגים את תחילת, סוף ופלט הקרייה בהתאם.

The New England Journal of Medicine is a registered trademark of [QA\("Who is the publisher of The New England Journal of Medicine?"\) → Massachusetts Medical Society](#) the MMS.

Out of 1400 participants, 400 (or [Calculator\(400 / 1400\) → 0.29](#)) 29% passed the test.

The name derives from "la tortuga", the Spanish word for [MT\("tortuga"\) → turtle](#) turtle.

The Brown Act is California's law [WikiSearch\("Brown Act"\) → The Ralph M. Brown Act is an act of the California State Legislature that guarantees the public's right to attend and participate in meetings of local legislative bodies.](#) that requires legislative bodies, like city councils, to hold their meetings open to the public.

איור 7 מתוך מאמר [2] – דוגמאות לחיזוי של המודל ToolFormer, הטקסט הצבעוני בסוגרים מרובעים אינם נראה למשתמש, אלה נמצא על מנת להורות למודל להשתמש בכלי חיצוני. ניתן לראות את השימוש בפלט הכללי משולב בהמשך הרצף.

העובדה שחל שיפור ביצועי מודל השפה כתוצאה משילובו עם כלים חיצוניים אינטואיטיבית בהחלט, בהמשך נראה את ההיגיון המתמטי העומד מאחורי שיטה זו.

עד כה הכרנו את פועלתו של מודל ToolFormer בפשט וכעת נעבור לדון בעומק הארכיטקטורה, תהליכי הכנת הנתונים, האימון וההסתקה.

שלב 1: הכנות נתונים

אחת מן הברקות במאמר [2] נמצאת בשלב הכנות הנתונים והוא הניצול של תוכנות הלמידה התרבותית (learning-in-context) של מודל השפה להכנות סט נתונים הלמידה. בהינתן מאגר מידע טקסטואלי X המשמש לאימון של מודל שפה רגיל, נפיק ממנו את המאגר הערוך X' המכיל את קטעי הטקסט המקוריים בתוספת הסמלים המיוחדים המורים על קריאת API, סוגה ואת הקלט והפלט של הקרייה (דוגמאות באיור 7).

הכנת המאגר X מורכבת מהשלבים הבאים:

הגימת קרייאות API – מנוסחת שאליתה למודל שפה והיל בבקשת להוכיח דוגמאות לטקסט משולב הוראות API בהינתן מספר דוגמאות לאופן שבו המודל נדרש להוכיחם. זהו השלב שמאפשר לאמן את מודל *ToolFormer* מוביל השקעה אידירה של בני אדם בהכנות דוגמאות ע"י ניצול יכולות הלמידה התוך-הקשרית של מודל שפה. לפלט יופיע הטקסט המקורי משולב עם הקרייאות API יתכן ריבוי של קרייאות עברו סמל יחיד ברכף.

Your task is to add calls to a Question Answering API to a piece of text. The questions should help you get information required to complete the text. You can call the API by writing "[QA(question)]" where "question" is the question you want to ask. Here are some examples of API calls:

Input: Joe Biden was born in Scranton, Pennsylvania.

Output: Joe Biden was born in [QA("Where was Joe Biden born?")] Scranton, [QA("In which state is Scranton?")] Pennsylvania.

Input: Coca-Cola, or Coke, is a carbonated soft drink manufactured by the Coca-Cola Company.

Output: Coca-Cola, or [QA("What other name is Coca-Cola known by?")] Coke, is a carbonated soft drink manufactured by [QA("Who manufactures Coca-Cola?")] the Coca-Cola Company.

Input: x

Output:

איור 8 – דוגמא לשאלת הכנת דוגמאות לקרייאות API, מתוך מאמר [2]

ביצוע הקריאה לAPI

כמשמעות לאחר שנאספו קבוצה של קרייאות API עברו כל סמל מתאים, מתבצעת הקריאה אליהם והתוכאה משתלבת בתחום הסמלים המיוחדים לאחר סמל החץ ">-" (אייר 7).

סינון הקרייאות

כעת נזדוד את אפקטיביות התוצאות בחרדת **האנתרופיה המשולבת של המשך חיזוי הטקסט**, וננסן את כל הקרייאות שלא עומדת בסף סינון τ_f (filtering threshold). עברו רצף תווים נתון נסמן את הקריאה המועמדת לסינון מס' i בثور: $(r_i, c_i) \in e$ כאשר c_i זוג סדור עם מידע על שם הכלי (מחשבון, תאሪיך...) והקלט לקריאה, r_i הפלט של הקריאה. M מודל השפה, w_i פרמטרים נלמדים (משקולות) ו- P_M פונקציית הסתברות מותנה.

נשתמש בפונקציית אנתרופיה צולבת ממושקלת הבאה:

$$L_i(\mathbf{z}) = - \sum_{j=i}^n w_{j-i} \cdot \log P_M(x_j | \mathbf{z}, x_{1:j-1})$$

למען חישוב *loss* במצבים בהם הקריאה מתבצעת והפלט משולבת ברכף, הקריאה לא התחכעה (רק הסמלים של הקריאה מופיעים, ללא הפלט מהקריאה) ובולם הקריאה לא שולבה

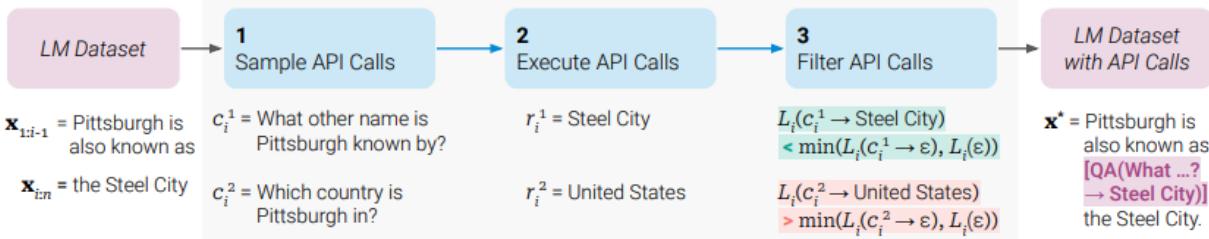
כלל בתקסט. משתמשים במופעים הבאים של פונקציית $loss$ לקבלת אומדן שיפור חייזי הרצף ע"י קראיה e :

$$L_i^+ = L_i(e(c_i, r_i)), \quad L_i^- = \min(L_i(\varepsilon), L_i(e(c_i, \varepsilon)))$$

בහינתן ומתקיים התנאי:

$$L_i - L_i^+ \geq \tau_f$$

הקריאה (e, c_i, r_i) תוגדר בתור עיליה ותשולב בתור המאגר כסיפה של רצף התווים הרלוונטיים, ראו איור 9 הבא המסכם את שלב הכתת הנתונים.



איור 9 סיכום שלבי הכתת הנתונים מתוך מאמר [2]

לסיכום, מאגר המידע המתkeletal * א מכיל את המאגר המקורי בתוספת רצפים של קריאות לכליים חיצוניים אשר נבדקו להטיב את איות הפלט.

שלב 2: האימון (training)

נכצע טיב המודל (fine tuning) עד לקבלת דיקוי רצוי ויתקבל מודל שפה M אשר מחליט באופן עצמאי מתי ואיזה כלי לשלב בתשובה בזמן חילול רצף הפלט. מכיוון שהקריאות במאגר האימון המעודכן נבחרו לפי תרומותם לדיקוי חייזי המשך הרצף, המודל למד (בשאיפה) באיזו עת ברצף הפלט קריאה לכלי חיצוני תשפר את תשובתו.

שלב 3: ההסעה (inference)

בתהליכי הסעה המודל M מחולל רצף שגרתי עד לקבלת הסמל המיוחד ">-". בו עוצר את פעולתו ימישיר לחולל רק לאחר קבלת תשובה מהכלי החיצוני אליו פנה כאשר פלט הכלי משולב במנגנון הקשב של המודל.

לסיכום, ToolFormer פותר את בעיית ההזיה במודלי שפה, ואף יותר מזה, משפר את הביצועים הכלליים של המודל. בעזרה לימון מונחה עצמי (self-supervised) עם התערבות מינימלית של בן אדם לומד המודל להשתמש בכלים חיצוניים ולשלב את התוצאה שלהם בפלט שלו, בכך משפר את איות התוצאה, בפתרון בעיות אРИתמטיות ומענה על שאלות המשתמש מבליל לගראע מיכולות השפה הבסיסית של המודל המקורי.

כעת נתבונן בפרשפקטיביה נוספת של כל' חיצוני העוזר בפתרון בעיות סימבוליות, אלגוריתמיות ומתחממיות ע"י המרת הכעה לשפה טבעית לקוד אותו מרים המודול ומשלב את פלט הריצה בתשובה לשאלתה.

2.2.3 מפרש קוד – Code Interpreter

ניקח את המודול שלב אחד קדימה בדרך להפיכתו לסקון – נוסיף לו את יכולת לכתוב ולהריץ קוד ולשלב את פלט ההריצה בתשובה למשתמש. יכולת זו מקדמת מספר אינטראטים בשיפור יכולות המודול, בהם:

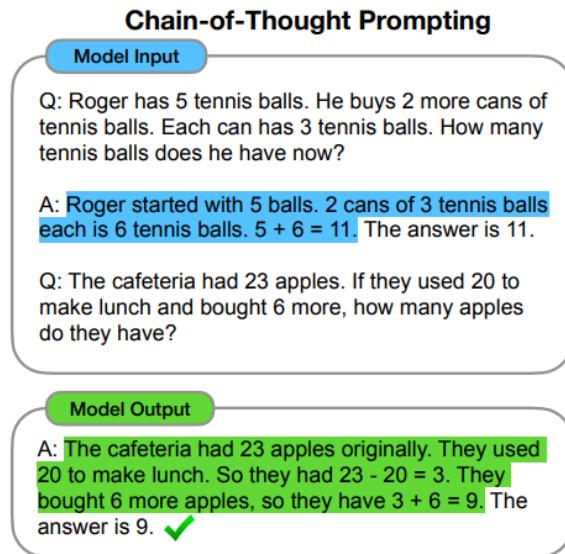
1. שיפור יכולת פתרון הבעיות של המודול
2. ייצור כלים ייעודיים לשימוש עתידי חזר
3. התמצאות עם כלים חיצוניים שלא אומן עליהם מראש

מעבר לקידום יכולות המודול לתת מענה איקוטי לשאלות המשמש, פעולה אלה מפתחות את **העצמאות** של המודול לבצע משימות מורכבות יותר ללא הסתمرا על הנחיה מפורשת מהמשמש.

לפנינו שנדגים את מפרשוי הקוד נציג רעיון שעוזר לנו את יכולות מודל השפה באמצעות שאלות שאותן שומרוכבת מדוגמאות של פתרונות לבבעיות דומות לאחת הנתונה ושםו few shot prompting.

הנחיה מעושרת-דוגמאות – Few shot prompting

בשיטת הנחיה זו מציגים למודול מספר קטן של דוגמאות קלט-פלט בתוך השאלה המטרת להכוין את המודול לפלט הרצוי ללא אימון חדש או כוונון עדין (fine tuning) שלו. השיטה מאפשרת למודל לנוף את יכולותיו החבויות בין מיליארדי הפרמטרים של שנלמדו בשלב האימון.



איור 10 – הינה מעוشرת דוגמאות כאשר המקלט המסומן בכחול הינה הדגמה לאופן החשיבה המדרש מהמודל (מתוך מאמר [4])

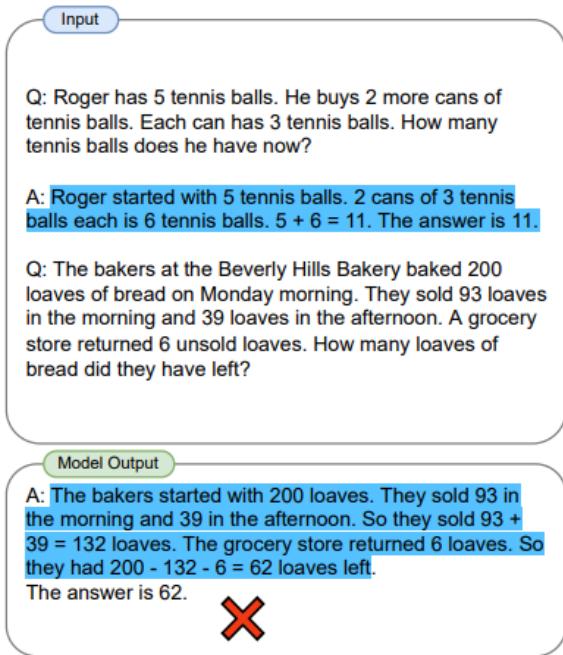
השו את אייר 10 לאייר 6 מוקדם יותר בסטינר, ניתן לראות שמודול השפה הצליח לפתור בעיה אריתמטית לאחר שכשל בה כאשר סופקה לו הנחה נטולת תיאור אופי החשיבה הנדרש. שיטה זו נקראת שרשרת מחשבה (CoT – Chain of Thought) והצלחתה יזועה בתחום מינוף יכולות המודל, כפי שהצליחה להשווות מודלים לבייצועי state of the art בזמן הכתיבה (2023).

שיטות רכבות המשלבות הנחה מעוشرת התפרנסמו בעקבות המאמר [4] לעיל, באחת מהן נדון מיד ובזאת נחזור לנושא הנוכחי **"מפרש קוד"**.

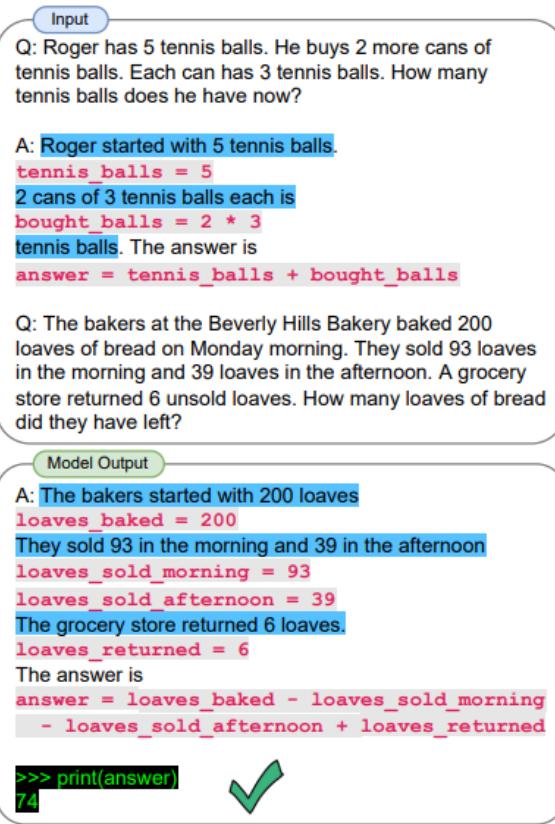
במאמר [3] מוצגת שיטת המודל הנתרמן תכניות (PAL – Programming Aided LM), אשר בהינתן הנחה מעוشرת-דוגמאות כתובת את שלבי הבניים לפתרון הבעיה בתור **קוד פיתוח**. לאחר פריסת כל שלבי הפתרון מרים המודל את המפרש ומציגו למשתמש את פלט התוכנית בתור התשובה הסופית.

כמעט ולא תיתכן טעות חישובית עבור קוד שנכתב בצורה תקינה (למעט זליגת זיכרון או בעיות בזמן ריצה), لكن בהנחה Sh-PAL הפיק שלבי בניים נכונים לפתרון הבעיה תשובה המפרש להנחה גם כן תהיה נכונה. לא ניתן לטעון כך עבור מודל CoC כיון שאפילו אם פירק את הבעיה באופן תקין, המנווע המילולי של מודול השפה עלול לטעת בחיזוי התשובה הנcona – הוא אינו מודijk כמו מפרש שפות התכניות.

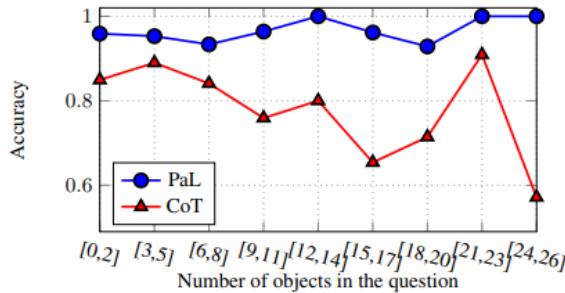
Chain-of-Thought (Wei et al., 2022)



Program-aided Language models (this work)



איור 11 – PAL פותר בעיה בהצלחה לuemת CoT הנכשל בה (מתוך [4])



איור 12 – ביצועי PAL גוברים על CoT עבור בעיות במורכבות הולכת וגוברת באמת מדיה (benchmark) של זיהוי אובייקטים צבעוניים ומענה על שאלות סקרה לגבים (מתוך [4])

מアイורים 11 ו-12 נסיק שיכולת חישובית של מודל שפה משתפרת ממשמעותית כאשר משלבים שרשרת מחשבה CoT עם מודל נתמך תכנות PAL. הרעיון הראשוני תומך בבנייה ההנחיה והשניי במקורה חזק של פתרונה.

שילוב זה אינו אלה הרמונייה הנובעת מעוצמת מודל השפה להבין כוונות והקשר תוך-טקסטואלי וכתוכאה פירוק הנחיה לתח-בעיות, והיכולת של שפות תכנות לפטור בעיות סימבוליות (אריתמטיקה, הגיון ואלגוריתמיקה).

סיכון בינויים

בחלק זה הצגנו את הכלים בהם נעזר מודל שפה גדול למען שיפור דיק הפלט, ובכך השתפר בפתרון בעיות שבahn דיק אրיתמטי ולוגי חשובות. מודל השפה נלקח צעד אחד קדימה המשע הפיכתו לסוקן במיוחד בעית כשבועית ההזיה פחתה משמעותית.

עתណז בעומקית יכולת החשיבה של המודל ובכך נשפר עוד יותר את האוטונומיות שלו במשמעות על הנחיות מורכבות יותר.

2.3 חשיבה عمוקה

עד למשפט זה בסמינר מודל השפה שלנו כבר יכולות Ai-cotiyot שփרו את ביצועי משמעותית ביחס למודל השפה נטול הכלים. באמצעות מודלי עזר פשוטים ([כלים חיצוניים](#)) הוגכלה יכולת הדיק שלן לצד צמצום בעיית ההזיה ועל ידי [מפרש הקוד](#) פוטר בעיות סימבוליות טוב יותר. כאמור, עד כה נספו למודל השפה יכולות חיצונית – ובהחלה התוצאות מרשימות: המודלicut מסוגל לא רק לפולוט טקסט Ai-cotiyot ומודיק, אלא גם לבצע פעולות בשמו.

אר היכולות שאספנו עד כה, עם כל הרכיבים אינן מספקות בכך לשמש לסיכון עצמאי. המודלicut לא מסוגל לבצע את המשימה שהגדכנו לחילק המידע של סמיןר זה: [פיתוח מערכת מסחרalgorythmiyim matbe'ot kripotografim](#), הטומנת בתוכה מורכבות רבה וידע במגוון תחומים כמו ידע בטכנייקות מסחר, עיצוב מוצר, ארכיטקטורת web, פיתוח הקוד ותיעודו ועוד.

כעת נרצה להעшир את המודל ביכולות עצמאיות, והעיקרית שבהן היא יכולת החשיבה העמוקה. בחלק זה נעסק ביכולת מודל השפה לפרק הנחיה מרכיבת של המשמש **למחשבות** ביןימים כך שכל מחשبة תורמת לדיקוק הפלט הסופי. נציג התפתחות היסטוריית קצרה של גישות לחשיבה عمוקה ואת תרומתן לעצמאות הכללית של המודל (ובהמשך, הסוכן).

להלן מספר גישות שנכנו אחת מעל השניה וכל אחת בתורה קידמה את יכולת החשיבה העמוקה של מודל השפה.

2.3.1 שרשרת מחשבה (Chain of Thought – CoT)

כפי שהוזגה בפרק [few shot prompting](#) היא טכניקה לכוון את מודל השפה לפרק את החשיבה לשאלבים בהינתן דוגמאות מוליכות בהנחיה.

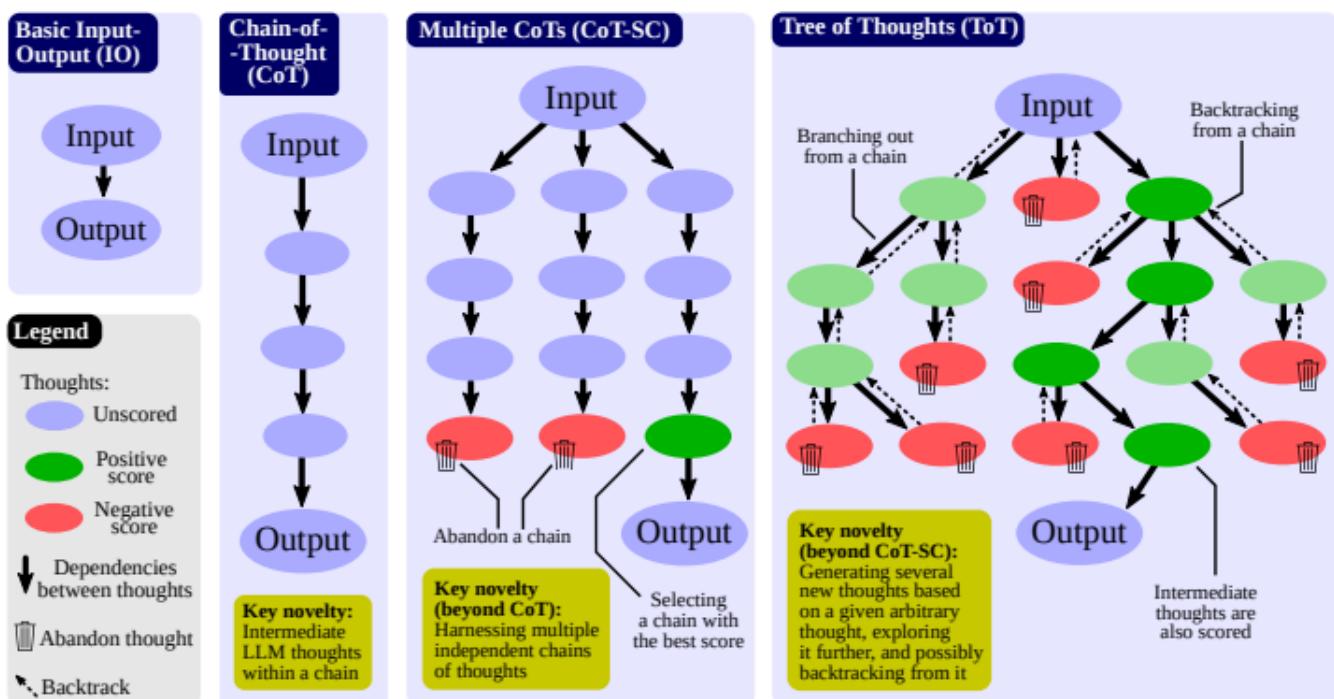
על טכניקה נבנתה שיטה בשם שרשרת מחשבה עם עקביות עצמית CoT-SC (Chain of Thought Self Consistency) המחוללת מספר שרשרות מחשبة כמו בCoT ובפלט נבחנת התשובה הגדירה ביותר. הצלחה שלה על פני שרשרת מחשبة בודדת מיוחסת לעובדה שלכויות חשיבה קיימות דרכים לפתרון והפתרון הוא זהה בכל הדרכים השונות, אך הפתרון הנפוץ ביותר בין כל שרשרות המחשכה הוא הסביר להיות נכון.

לא נפרט על ארכיטקטורת הפתרון של שיטות אלו, ובמקום נמקד את החרחבה על שני השיטות המודרניות הבאות: עץ מחשבה וגרף מחשבה.

2.3.2 עץ מחשבה (ToT – Tree of Thoughts)

במאמר [5] מתווכחים החוקרים שהשיטות הקודמות לעידוד חשיבה רב-שלבית של מודל שפה לא תואמת את החשיבה האנושית. יכולת החשיבה העמוקה של מודל שפה סטנדרטי (או שימוש בהנחהיות CoT) מוגבלת על ידי האופן הרציף בו הוא מחולל את הפלט – סמל אחר סמל שהוא שרשרת סתובירות סדרתית, בעוד שהחשיבה האנושית דומה יותר לחיפוש למרחב בעיה קומבינטורית ככלمر חיפוש בעץ בו הצמתים מייצגים פתרונות חלקיים והקשות פועלות המשנות את הפתרון החלקי בכיוון הפתרון.

פתרון המוצע במאמר הוא אלגוריתם חיפוש בעץ, כאשר החזקה שלו מתחבطة באופן חילול "מחשבות ביןימ" (הצמתים) והעריכתם ע"י מודל השפה עצמו. בכל שלב נבחרים עד t צמתים (במאמר $3 \leq t$) שהוערכו כציונים הגבוהים ביותר על ידי המודל ואליהם מחבר האלגוריתם עד k מחשבות **שונות** חדשות (במאמר $5 \leq k$) אשר חילל מודל השפה, ורק חוזר חיללה עד לפט.



אייר 13 – השוואת בין פרדיגמות חשיבה عمוקה [6]

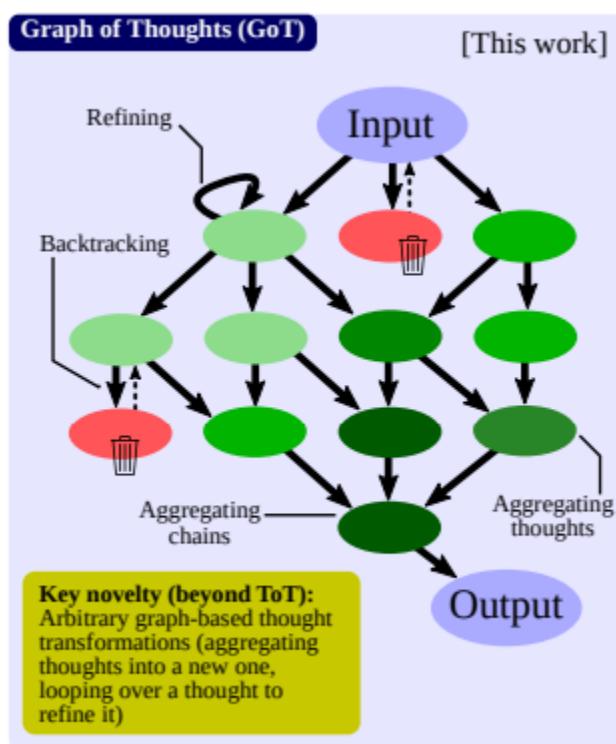
שיטת זו חזקה מקודמותיה בשני אСПקטים:

1. השיטות הקודמות (בעיקר CoT-SC) לא מחברות מחשבות בין ענפים שונים בעץ ובכך מביאות פוטנציאלי לשלב איכיות נבדלות משנה שרשראות מחשבה שונות.
2. ToT מקיים אלמנט של תכנון, חזרה לאחר (backtracking), הערכה וסינון מחשבות ועל ידי כך מאפשרת חקירה מיטבית של מרחב הפתרונות.

חשיבות לציון כי בשונה מקודמותיה, שיטה זו משתמש במודול השפה בתור מודול (*module*) באlgorigthm כאשר הפניות אליו מתבצעות מתוך אחת הדרישות: חילול מחשבה והערכת מחשבה. עז המחשבה נבנה ונשמר בזיכרון חיצוני למודול על ידי ריצה של קוד האלגוריתם.

2.3.3 גרפ' מחשבה (Graph of Thought- GoT)

כעת נציג גישה חזקה יותר בשם "גרף מחשבה". החוקרים שעומדים מאחוריו פיתוחה מתווים כיעודם על עליונות הביצועית שלה הנובעת מtower מודול גרפי דינמי שרירותי הדומה ביותר לאופן החשיבה האנושית, על ידי הסרת המגבילות ההדוקות של מבנה העץ או השרשת. גישת גרפ' המחשבה נבנית יישירות מעל הגישות שהוזכרו לעיל (ToT, CoT-SC) ומרחיבה אותן ע"י הוספת גמישות לתהליכי התפתחות החשיבה העמוקה וכתוצאה מושפרת את ביצועי המודול תוך שימוש מופחת במשאבים – מאפשרת דפוסי חשיבה מורכבים יותר.



איור 14 – תיאור חזותי של GoT, בהמשך לאייר 13 [6]

הגרף משתמש במודול השפה בתור רכיב חיצוני ומבצע אליו קריאות במהלך זמן הריצה.

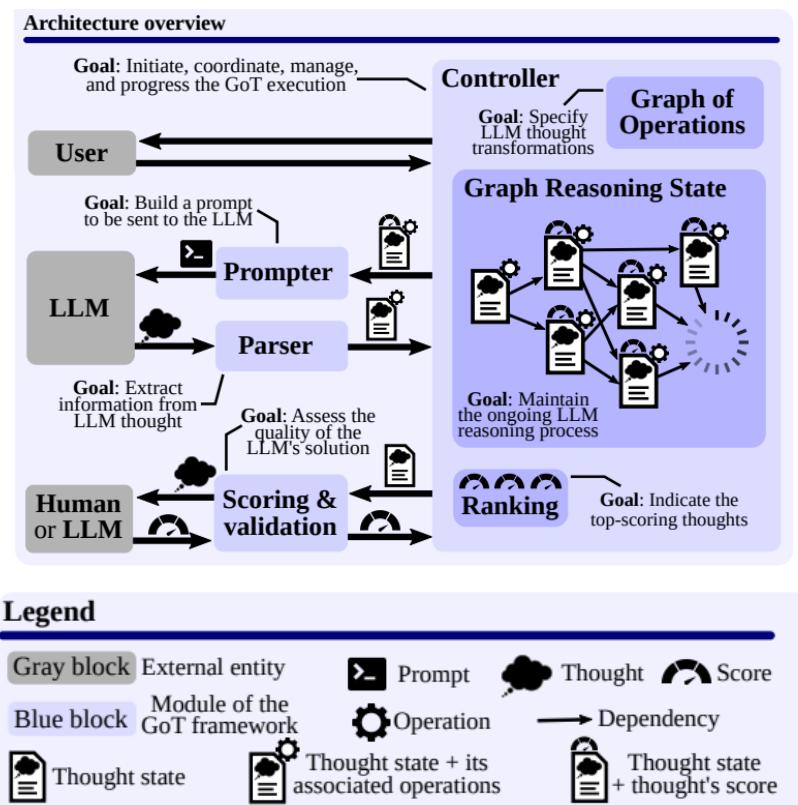
ההידושים של גישת הגרף על פני עצים מתבטאת במספר אופנים חשוביים המקרבים אותה לעליונות ביצועית. תחילתה נשים לב שהגרף מאפשר לקשותות לחבר צומת לעצמו וכן מאפשר השבחה של מחשבות קיימות. שנית, מחשבות שהובילו לעלה (מחשבה סופית\פלט) שאנו עלולות לתרום לרצף מחשבות אחר שבכו לא השתתפה כלומר שרשרות מחשבה נבדלות משתפות מחשבות ובכך מועשר מרחב הפתרונות. לבסוף, מתנקזות כל שרשרות המחשבה לעלה בודד וסופי, המרכז את כל המחשבות התורמות ומצקק אותן לכדי פתרון סופי יחיד.

ארQUITקטורת המודול משלבת מספר מרכיבים שפעולותם המשותפת מנננת את יכולות מודול השפה. כמו בשיטת העץ, גם

מרכבי הארץQUITקטורה הם:

1. Controller – מנצח (Controller) – מנצח (Orchestrates) את תהליכי החשיבה. עוקב אחר גרפ' הפעולות (OGo), בוחר אילו מחשבות לקדם, בוחר באופן הפעולה (האם לייצר מחשבה חדשה, לרכז מספר מחשבות, לשפר מחשבה וכו'). מחליט האם לעזר את הפעולה של גרפ'

- המחשבות בהגעהו למחשבה סופית מספקת או להמשיך. מטרתו לקדם את גרפ' המחשבות לכיוון הפתרון באופן יעיל.
2. גרפ' הפעולות (GoO – Graph of Operations) – מייצג את הליך המחשבה הכללי חד פעמי וסיטוי מתחילה ריצת מודל הגרף. מגדר את אופן פירוק בעיה לגורמים ואינוחם למען הפתרון.
 3. Promter – מייצר הנחיות למודל השפה בהתאם לבני הגרף הנוכחי באמצעות שורת פקודות. באחריותו להנחות את מודל השפה לייצר מחשבות חדשות, לשפר מחשבות קיימות, לוודא תקינות או לרכיב מספר מחשבות.
 4. Parser – מחלץ מידע מבני מהפלט הטקסטואלי של מודל השפה ומיציר מצב של מחשבה. מעדכן את מצב הנמקת הגרף (GRS – Graph Reasoning State).
 5. דירוג ואיימות (Scoring and Validation) – מעריך את הנכונות והaicות של כל מחשבה בעזרת מספר כלים שונים כמו פונקציה מקומית שהוגדרה מראש, היעזרות במודל השפה או בגין אדם. תוצאה דירוג המחשבה תתמוך בהחלטות Controller לגביה חידוד או הזנה של מחשבות קיימות.
 6. מצב הנמקת GRS – מתחזק מצב דינמי של החשיבה (או הנמקה). מייצג במבנה נתונים את כל המחשבות שחוללו עם דירוג ואיימותיהן ובנוסף עוקב אחר משימות גרפ' פעולה GoO שבוצעו. מציג את ההקשר הנוכחי לביצוע החלטות מודל הגרף.



סיכון בינוי

הציגנו מספר גישות הממנפנות את מודל השפה לפתרון מגוון רחב של בעיות בתחוםים שונים וביניהם כתיבה יצירתיות, בעיות הגיון, ארכיטקטורת אлогריתמית ועוד. עובדה אחת הבולטת בהתחזיות הלו היא שלב对她ותה פתרונות חשיבה עמוקה עובר בהדרגה מודל השפה לתפקיד בתור רכיב חיצוני במערכת גדולה יותר. בCoMo מודל השפה היה הרכיב היחיד בפתרון אך בזמן שהתרנס מאמר HoGo כבר בכירור סוכם הקונצנזוס האקדמי שמודל השפה בתור חבילת תוכנה נפרדת המשולבת בפתרון עם ארכיטקטורה מורכבת מניב תוצאות אינטראקטיביות יותר.

נוסף על ההיפרדות הלוגית – יכולות חילול המחשבה, הסקת מסקנות (התמרת מחשבה), דירוג מחשבה והערכת אינטראקטיביות בולטות בתור בתחוםי האחריות של מודל השפה.

וכעת הגיעו לפסגת בינויים של הסמינר. דיברנו על [מודל השפה הגדול](#) ואופן פועלתו, חיברנו למודל [כלים חיצוניים](#) בהם מוצע שימוש לשיפור תוצאות והרחבת יכולות ולבסוף על יכולות [המחשבה העמוקה](#) שלו – שלושת התנאים החכרתיים לבנייתו של [סוכן מודל שפה](#).

עתה כשברשותנו א芝ה והבנה במונח הסוכן, נדוזן במערכות מרובות סוכנים ונשאל ונענה על שאלת התועלת בריבוי סוכנים בתוך מערכת ייעודית לפתרון בעיות מורכבות.

2.4 מערכת מרובת סוכנים (Multi-Agent System)

2.4.1 מוטיבציה

מאז שחר ההיסטוריה תועדו בני אדם ובعال' חיים אחרים שמשתפים פעולה למען השגתיעד גדול מכדי שמשתף יחיד, בין אם זו קבוצת לוחמים של שבט ציידים-לקטים בצד אחר ממوطה או קבוצת מהנדסי תוכנה ומנהלי מוצר מSchedulerים פיציר חדש ב מוצר של החברה. הדברים, המאפשרות על ידי גוףزر להשמדת מתחדות לנחיל ויצאות למתקפה משותפת להגנת הקן. מאותן סיבות בדיקת כמה המוטיבציה לקבץ מספר סוכנים לכדי מערכת אחת ובכך למנף את התפקיד המשותפת שלהם.

סוכן מודול שפה גדול בעל יכולות מגוונות ורבות, אך לעיתים תוכנה רצiosa יותר היא מקצועיות מרבית בתחום צר ופסיכי. כל מודול שפה יכול לכתוב קוד, שירה ולפצח אתגר ארכיטקטוני באוטה הנחיה יחידה – אך לרוב אין בקשר צורך. בעולם האומית, נעדיף שהסוכן אחד יהיה הכى טוב בכתיבת קוד, השני בארכיטקטורת מוצר והשלישי בחשיבות עסקית. שייתופ פעולה בין מומחים בין שיכלה למען מטרת משותפת אחת היא שם המשחק וזוהי המוטיבציה הראשונה מאחוריו מערכות מרובות סוכנים – **מומחיות**.

מוטיבציה נוספת הנובעת כמעט מיידית מהפרדת סוכני מומחיות הינה **דרגיות (scalability)**, הרि כאשר קיים מומחה בתחום מסוים הוא יכול לספק שירות לכל "לקוח" שזוקק בו, בדומה ליעצים בעולם הקפיטליסטי המודרני.

גיאון מחשבתי הנובע משייתוף פעולה של מומחים הוא עוד מוטיבציה מתבקשת מערכות מרובות סוכנים.

2.4.2 ארכיטקטורות

ישנן שלוש גישות מרכזיות בתכנון ארכיטקטורות של מערכות סוכנים: שליטה מרכזית (centralized), שליטה מכוורת (peer-to-peer) ומודול היברידי. להלן תיאור ו יתרונות וחסרונות מרכזיים של כל גישה:

1. שליטה מרכזית – בקר ראשי (תוכנה, מודול שפה או אדם) שמנהל את תהליכי המחשבה על ידי פירוק המשימה הראשית לתת משימות וחולוקן לסוכנים המתאימים במערכת. יתרון הוא תהליכי העבודה מפוקח ומאורגן, סוכן ייעודי לניהול התהליכי. החסרונות שסוכן ניהול גבוה משאבים נוספים ועלול להגדיל את מרחיב הדיזון בעקבות פיקוח חזק.

2. שליטה מכוורת – הסוכנים מתחברים בינם לעצם ללא בקר מרכזי. היתרון הוא שסגן זה מדמה קבוצת מומחים המתדיינת בנושא באופן פתוח ומוגן ובכך מאפשר CISI'י רחוב יותר של מרחב הבעיה. החיסרון הוא אתגר ניהול המשימות בעקבות העדר סוכן ניהול.

3. היברידית – שילוב של שני הגישות לעיל, מכפרת על החסרונות של שניהם במחיר של מרכיבות ארכיטקטוניות.

עד כאן התדינו לעומק במרכיבים של הסוכן – מודל שפה גדוֹל, כליים חיצוניים וחשיבה عمוקה. בנוסף ציינו את המוטיבציה ומספר ארכיטקטורות למערכות מרחבות סוכנים וcutet ברשותנו ההבנה הנדרשת על מנת להמשיך לחלק המעשי בו נהנדס ונפעיל מערכת מרחבות סוכנים לבניית תוכנה הסוחרת במטריות קרייפטוגרפים על ידי הפעלת אלגוריתמי סחר. נדון בהמשך ייצור הסוכנים והגדրתם, נפעיל אותם ונעריך את איכות התוצאה הסופית.

3. חלק מעשי: הדגמה – פיתוח מערכת מסחר אלגוריתמי במטבעות קרייפטוגרפיים באמצעות מערכת מרובת סוכנים

3.1 מבוא

מטרתו של הפרויקט הייתה התנסות בהפעלה של מערכת מרובת סוכנים לבניית מערכת תוכנה מורכבת – מערכת למסחר אלגוריתמי במטבעות קרייפטוגרפיים ואת הכלים החזותיים להצגת נתונים כמו שער מטבעות, סטטיסטיות, התממשקות מול API צד שלישי ופיתוח אלגוריתם לקבלת החלטות רכישה ומכירה. בפועל, ניסיונות רבים בהפעלת מערכות מרובות סוכנים ([MetaGPT](#) – open source multi-agent framework, [MGX](#) – commercial multi-agent system) נכשלו לבניית המערכת. לאחר התנסות במספר גישות וכליים שונים, נבנה רק חלק מטור המערכת השלמה.

בחילק זה אציג את המערכת הבנوية ואת התהיליך שהוביל לבנייתה, את הכלים השונים שהשתמשתי, את הגישות מאחוריהן, התוצאות והמסקנות.

3.2 המערכת AlgoTrader

מטרת המערכת היא לסמן לsonianות רכישה ומכירה חזקות על מנת ליעל (או להחלף) את הסחר האנושי במטבעות. למעשה, בהינתן שינוי קטן בkode והוספה גישה למשתמש binance, המערכת תבצע פעולה מסחר עם ערך מטבע אמיתי המוחזקם בארכך של המשתמש.

המערכת בנוייה מהמרכיבים הבאים:

צד השרת (server side)

1. התממשקות עם API WebSocket של פלטפורמת המסחר Binance.com ובכך מתאפשרת קבלת נתונים מסחר בזמן אמת ונתונים היסטוריים
2. מנוע קבלת החלטות לרכישה ומכירה, אסטרטגיה קלאסית EMA Crossover
3. התממשקות זרימת נתונים בזמן אמת (socket) עם הצד הלוקאלי

צד לקוח (client side)

1. עמוד המציג את נתונים המסחר, גרפים של מחיר המטבע והMOVING AVERAGE הנעים ואינדיקטורים על נפח מסחר, כמות עסקאות ועוד

2. התחברות לשרת באמצעות sockets



איור 16 – צילום מסך של משקען מערכת המסחר. מוצגים גרפים של מטבעות BTC ו-ETH, עם נתונים המחיר בשוק, ממוצעים נועם באורך 21 ו-55 העומדים בבסיס החלטות של אסטרטגיית המסחר בשם EMA Crossover וחצים ירוקים ואדומים המסמנים הוראות קנייה וממכירה בהתאם.

הקוד ותיעודו נכתבו במלואם על ידי סוכני מודול שפה גודלה בהינתן הנחיות והגדירות. הקוד זמין לצפייה הורדה והרצה באתר הגיט האב שלו – <https://github.com/boriboy/algotrader>

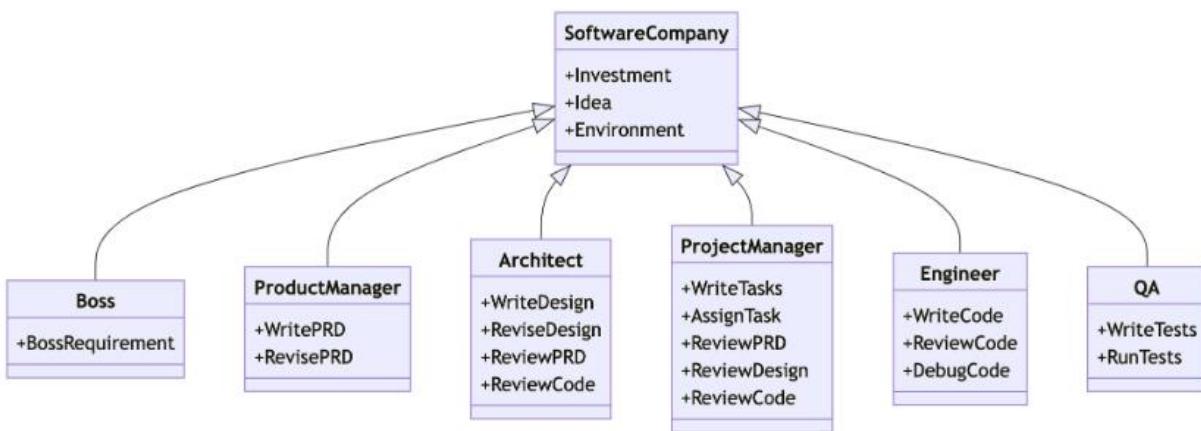
3.3 תהליכי פיתוח וגישה שונות למערכות מרובות סוכנים

כפי שצוין לעיל החזון למערכת המקורית התכווץ בעקבות מספר ניסיונות כשלים של הפעלתם של שביכות מרובות סוכנים, כאשר אותה מערכת הוגדרה להיות גודלה יותר בנפח ובMORECBBOT ההנדסית, ביןיהם ריבוי מספר אסטרטגיות מסחר שונות ומגוונן הצבעה שכיח ממנפים את רוחניות המסחר, שליטה משתמש מוגברת על פעולות המערכת והתממשקות מלאה למשתמש סוחר ב-Binance על ידי ביצוע עסקאות מסחר.

למרות שלא השתמש הרעיון המקורי, ההתנסות עם הכלים השונים לימדו אותו לא מעט על עובודה מול סוכני פיתוח תוכנה ואת הידע אסכם בסעיף זה.

ניסיון 1 – מערכת מרובת סוכנים קוד פתוח MetaGPT

בניסיון הראשון השתמשתי בכל קוד-פתוח בשם [MetaGPT](#) שפותחה ומותחזק על ידי מספר חוקרי מודל שפה ולמזהה عمוקה, ביחד הם יצרו מערכת שבහינתן הנחיה עם פירוט למערכת רצiosa מייצרת אותה עם תוכנה מלאה, מסמכי אפיון, ארכיטקטורה ואمنות דיגיטלי. העבודה קוראת מאחרוי הקלעים על ידי מספר סוכני מודל שפה גודל מוגדרם מראש לתקמידים שכחיהם בתעשייה ההי-טק: מנהל מוצר, ארכיטקט, מנהל טכנולוגיות, מהנדס תוכנה ומהנדס אבטחת אינטלקט. למעשה, כל ישות מתפקדת בתורו נוטן שירות שכביסו מודל שפה, כל הישויות יחד מנווהלות על ידי אלגוריתם איטרטיבי של ייצור משימות והשלמתן בתורות.



איור 17 – סכמת חברת פיתוח תוכנה, לפי הפילוסופיה של *MetaGPT* (מתוך קוד המקור)

לאחר הגדרת מפתח API של OpenAI והגבלת התקציב נדיבה של \$ 12 דולר אמריקאי ההנchia
שנמסרה למערכת היא זו:

Create an algorithmic trading software that uses Binance API to trade crypto currencies. The algorithmic trader should detect good buying and selling points aiming to make profit, using advanced trading strategies. Implement a simple web dashboard providing live insights on market data (drawn from binance via socket) and the algorithm's decision making. Use the binance's testnet sandbox environment for committing market instructions. Provide profit/loss stats on the dashboard, displaying the accumulated losses and profits along with other useful information on the algorithm's performance.

לאחר מספר דקות של ריצה פلتה המערכת קובץ הגדרת מוצר וככלה. כל הניסיונות להמשיך את הריצה או להתחילה כר שטים את עבودתה כשלו, עם חוב של \$ 4 המשכתי לניסיון הבא.

ניסיון 2 – מערכת מרובת סוכנים מסחרית MGX

מערכת זו מבוססת על תשתיית של הספרייה MetaGPT שאותה נזכרה לעיל ופועלת ענן בהפעלת הסוכנים דרך ממשק web ידידותי. בזמן הרצתה באותה הנקודה שצוטטה לעיל, עבדה המערכת מספר דקוט עד שהופיעה הודעה על סיום מוחלט של סיום בניית המערכת. כשבורת עלי הקבצים התבגר שחלק מן התיקיות ריקות ולא כל הקוד כתוב עד הסוף, מה שהפך את התוצר לבטלי ניתן להרצאה.

בשלב זהה חלה ההבנה כי ניתן אחד שני דברים: טכנולוגיות מרבות סוכנים לוקוט בשיטות ניהול פנימי או ההנחה שניתנה לה מרכיבת מדי. לפשט את ההנחה זה לא בא בחשבון מפני שמרכז ההtauניניות בסמינר זה הוא יכולת הביצוע של מערכת סוכנים שכזו, لكن הניסיון הבא בפיתוח המערכת הרצiosa כולל התערבות אנושית יותר אינטנסיבית.

ניסיון 3 – מערכת היברידית סוכנים ובן אנוש מפקח

בארქיטקטורת של מערכות מרובות סוכנים ציינה אפשרות היברידית בה קיים שילוב של ישות מנהה (סוכן או בן אדם) קבוצת סוכנים לצד קבוצה עצמאית. בסעיף זה חלה התערבות אנושית יותר משמעותית מאשר כתיבת ההנחה ולהזיכה על מקש ה"אנטר", **נלכמה האוטונומיה** של קבוצת הסוכנים.

כעת כאשר בן אדם הוא מנהל העבודה של הסוכנים, הוסרה אפשרות לתקלות סביב יכולת ניהול המשימות העצמי של קבוצת הסוכנים.

לעזרתי יצרתי 2 סוכנים מומחים המיעדים בתחום המומחיות שלהם:

1. מומחה פיננסי – נוצר בעזרת המודל 5-GPT ותפקידו לבצע מחקר מקורות אינטרנט בנוגע לשיטות (אסטרטגיות) מסחר במערכות קרייפטוגרפים ויצירת מסמך הגדרת שמיועד לקהל הטכני באופן שכל מהנדס תוכנה יוכל למש את הפונקציונליות של האסטרטגיה מקריאה במסמך.
2. מומחה להتمמשקות תוכנה עם Binance – גם מבוסס על מודל 5-GPT, הונחה לכתיבת קוד התממשקות (WebSocket, API) עם פלטפורמת המסחר על ידי שימוש במקבץ קבוע תיעוד מספרית התיעוד הרשמי של Binance שצורפו לו במאגר המידע

פייזור תחומי האחריות בין מספר סוכנים התגלתה כשיתה עילאה משמעותית יותר מהגדרת סוכן יחיד בעל תחומי אחריות רחב, אך התאפשר לכל מודל שפה לגלם תפקיד מוגדר היטב בתחום צר בעל עומק הבנה ואיכות ביצועית.

ניתן לצפות בתוצרים של הסוכנים המומחים בתיקייה המסמכים של הפרויקט [algotrader/docs](#)

ותוצרי הקוד של הסוכן המומחה להתחמשות עם Binance הורצו בהצלחה מהניסיון הראשון והם יהוו מסמכי בסיס למערכת הגדולה יותר – אוטה יפתח סוכן מודל שפה הידוע בהתחמשותו בכתיבת קוד, המודל 4 Claude Sonnet. לאחר מתן גישת קריאה וכתיבה לティקית הפרוייקט המקומית וסדרה של הנחיות לפיתוח ותיקוני באגים, יצר המודל את המערכת לעיל תוך כשבתיים של עבודה משותפת.

להלן רשימה חלקית של הנחיות שניתנו למודל:

1. fix the dashboard graphs to be less jumpy and more similar to those appearing on trading platforms.
2. when app is initiated, collect previous 50 minutes of symbol data for the ema crossover strategy to begin executing right away.
3. add ema information on top of the existing graphs.

Great. More changes:

1. fix the candlestick section in the graphs, it's all messed up.
2. Whenever `open_long()` or `close_position()` are evoked, display the decision at the appropriate place on the line graph as an arrow above the price graph.

אייר 18 – דוגמאות להנחיות למודל 4 בבנייה *Claude Sonnet* AlgoTrader

4. סיכום ומסקנות

טכנולוגית מודל השפה הגדול, העומדת בבסיסו של סוקן, עברה התפתחויות עצומות בשני העשורים האחרונים וזאת כתוצאה של מחקר אקדמי בתחום ה-NLP, התעצמות הכוח החישובי ושפע של מידע הזמן באינטרנט. מודל השפה מתבסש עם כלים חיצוניים ומשתמש בטכניות חשיבה عمיקה ובכך משתפרת משמעותית יכולת פתרון בעיות ומתחוסף מסוגלות לתפקיד אוטונומי – ל透זר זה קוראים סוקן. הסוקן-shellצממו שימושי מגוון רחוב של תחומים ביניהם התקשורת צ'אט (בידור, שירות לקוחות, עוזר אישי מנהל יומן ועוד), כתיבת קוד הרצתו ופענוח תקלות, תרגום בין שפות, כתיבה יצירתיות, סיכום טקסט וחיפוש באינטרנט.

מערכות מרוכבות סוכנים מספקות הרחבה לכליות הסוקן על ידי מינוף הפרדת תחומי אחריות והרשמת פרוטוקול תקשורת בין הסוכנים וניהול משימותיהם.

החלק המענייני היה בחלוקת בדיקה לרמת האוטונומיה הקיימת בכלים הפופולריים כולם במבנה מערכות תוכנה מורכבות. נקבע בזמן כתיבת סמיינר זה, רבים הפרסומים בפורומים ורשתות חברותיות באינטרנט המעידים על חשש אוכלוסיות העולם המתකם מפני אבדן העבודה לבינה מלאכותית. אמונה סמיינר זה אינה עוסקת בנושא החשש – חשוב לציין כי הפעלה "חסרת חשיבה" (prompt-to-complex software system) של סוכני מודל שפה והציפה לקבל מוצר תוכנה מורכב אינה מציאותית. יתרון כי מערכות פשוטות יותר אשר בנויות תועדה מספר רב של פעמים במארגני קוד פומבי תוך שימוש בספריות מוכראות ומתחודדות היבט תיינה קלות יותר להתמודדות עבור מערכות מרוכבות סוכנים, אך לייצר מוצר מורכב יצירתי הינה משימה לא טריויאלית.

סוכנים מתמודדים היבט עם משימות בהיקף קטן, כאשר יחד עם הכוונתו של האדם בהחלט ניתן להגיע לתוצר איקוטי. שימוש בסוכנים חוסך לאדם את הזמן המושקע בפתרון בעיות טכניות מצומצמות כמו כתיבת פונקציות קוד ואףלו מודולים שלמים, ביצוע מחקר בנושא ממוקד באינטרנט וסיכון ועוד. כמובן, ניתן לראות את סוקן מודל השפה בתור כלי **שמרחיב את יכולת והתפקוד של האדם**.

מקורות

1. Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).
2. Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. NeurIPS, 2024.
3. Gao, Luyu, et al. "Pal: Program-aided language models." *International Conference on Machine Learning*. PMLR, 2023.
4. Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. NeurIPS, 2022.
5. Yao, Shunyu, et al. "Tree of thoughts: Deliberate problem solving with large language models." *Advances in neural information processing systems* 36 (2023): 11809-11822.
6. Besta, Maciej, et al. "Graph of thoughts: Solving elaborate problems with large language models." *Proceedings of the AAAI conference on artificial intelligence*. Vol. 38. No. 16. 2024.