Automated Domain Modeling with Large Language Models: A Comparative Study

By Kua Chen et al.

Presented by Aviv Borodko & Noga Dines

הבעיה והמוטיבציה למחקר

הבעיה המרכזית

כיום אין דרך ליצור מודל תחום באופן אוטומטי לחלוטין ללא התערבות אנושית. שאלת המחקר המרכזית היא האם LLMs יכולים להפיק אוטומטית מודל תחום מלא מתיאור טקסטואלי יחיד, בלי fine-tuning ובלי עזרה אנושית?

שאלות מחקר נוספות

המחקר מתמקד בשלוש תתי שאלות: עד כמה מודלי שפה גדולים מצליחים במידול תחום? איך prompt engineering משפיע על הביצועים? איזה מודל שפה מציג את הביצועים הטובים ביותר?

המוטיבציה

כיום מידול תחום מתבצע ידנית על ידי מהנדסי תוכנה, תהליך הדורש זמן רב ורמת מומחיות גבוהה. הכלים הקיימים לאוטומציה דורשים התערבות אנושית או מתמקדים בניתוח משפטים בודדים ולא מבינים את התמונה השלמה.

הפתרון המוצע במחקר

גישה עיקרית: המאמר מציע לראות את מידול התחום כמשימת יצירת טקסט (Text Generation), ולא כמשימת סיווג או חילוץ מידע רגילה.

שיטת הפתרון:

הכנת פרומפט

ניסוח הנחיות ברורות ל-LLM, הכוללות תיאור המשימה, תיאור הבעיה, פורמט מדויק לפלט (לפי תחביר EBNF) ודוגמאות להמחשה (ב-N-shot/Zero-shot/CoT).

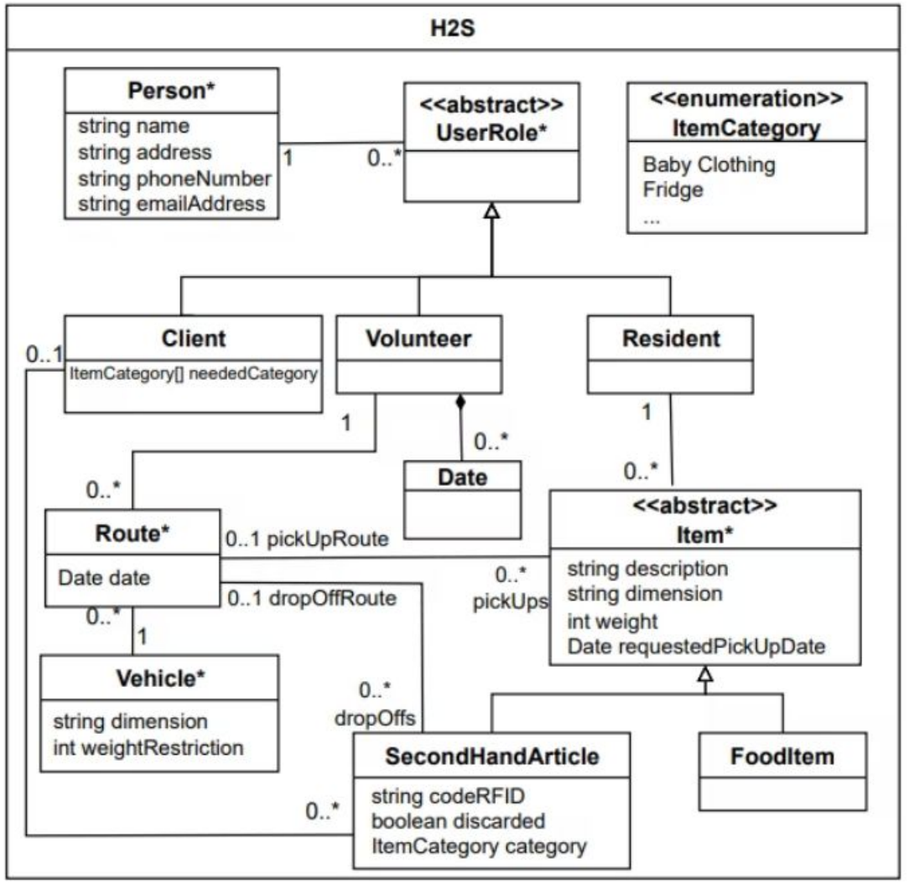
שליחה ל-LLM

העברת הפרומפט למודל השפה הגדול לעיבוד.

עיבוד אחרי קבלת התוצאה

ביצוע Post-processing להפקת מחלקות, מאפיינים וקשרים מהטקסט שהתקבל.

```
<class-diagram> ::= [<enumerations>] <classes> <relationships>
<enumerations> ::= "Enumerations: " (<enumeration>)+
<enumeration> ::= <string> "(" <literals> ")"
<literals> ::= <string> | <string> ", " <literals>
<classes> ::= "Classes: " (<class>)+
<class> ::= ["abstract"] <string>"(" [<attributes>] ")"
<attributes> ::= <attribute> | <attribute> ", " <attributes>
<attribute> ::= <type>["[]"] <string>
<relationships> ::= "Relationships: " [<composition>]* [<inheritance>]* [<association>]*
<composition> ::= <mul> <string> "contain" <mul> <string>
<inheritance> ::= <string> "inherit" <string>
<association> ::= <mul> <string> "associate" <mul> <string>
<type> ::= <string>
<mul> ::= "*" | <num> | <num>".."("*"|<num>)
```



Textual Domain Description: The Helping Hand Store (H2S) collects second hand articles and non-perishable foods from residents of the city and distributes them to those in need. H2S operates in many cities, but each location is run independently. To increase the number of items available for distribution, H2S is seeking to offer a Pickup and Delivery Service to its customers, which would allow a resident to schedule a pickup of items from a street address online at the H2S website.

A resident enters a name, street address, phone number, optional email address, as well as a description of the items to be picked up. H2S has a fleet of pickup vehicles, which it uses to collect items from residents. A pickup route for that day is determined for each vehicle for which a volunteer driver is available. Volunteer drivers indicate their available days on the H2S website. The route takes into account the available storage space of a vehicle and the dimensions and weights of scheduled items. A scheduled pickup may occur anytime between 8:00 and 14:00. After completing all scheduled pickups, the driver drops off all collected second hand articles at H2S's distribution center. Non-perishable foods, on the other hand, are directly dropped off at the food bank, which then deals with these items without further involvement from H2S ...

Enumerations:
ItemCategory(Baby_Clothing, Fridge, ...)
Classes:
H2S()
Person(string name, string address, string phoneNumber, string emailAddress)
abstract UserRole()
Client(ItemCategory[] neededCategory)
Volunteer()
Resident()
Date()
FoodItem()
SecondHandArticle(string codeRFID, boolean discarded,ItemCategory category)
Vehicle(string dimension, int weightRestriction)
abstract item(string description, string dimension, int weight, Date requestedPickUpDate)
Route(Date date)

Relationships:
1 H2S contain * Item
1 H2S contain * Vehicle
1 H2S contain * Route
1 H2S contain * UserRole
1 H2S contain * Person
1 Volunteer contain * Date
Client inherit UserRole
Volunteer inherit UserRole
Resident inherit UserRole
SecondHandArticle inherit Item
FoodItem inherit Item
1 Person associate * UserRole
0..1 Client associate * SecondHandArticle
1 Volunteer associate * Route
1 Vehicle associate * Route
0..1 Route associate * Item
0..1 Route associate * SecondHandArticle
1 Resident associate * Item

Fig. 1: Example domain model (left) and its textual representation (bottom right) with the problem description (top right)

Generate a class diagram...

Create a class diagram for the following description by giving the enumerations, classes, and relationships using the following format:

Enumerations: EnumerationName(literals)

Classes: ClassName(type attribute)

Relationships: mul1 Class1 associate mul2 Class2
Class1 inherit Class2
mul1 Class1 contain mul2 Class2

Description: The LabTracker software helps (i) doctors...

0-shot

Generate a class diagram...

Description: A city is using the Bus Transportation Management System...

Enumeration:
Shift(morning, afternoon, night) ...
Classes:
BTMS() ...
Relationships:
1 BTMS contain * BusVehicle ...

Repeat for N examples

Description: The LabTracker software helps (i) doctors...

N-shot

Generate a class diagram...

A resident enters a name, street address, phone number, optional email address, as well as a description of the items to be picked up.

-> Person(string name, string address, string phoneNumber, string emailAddress), abstract item(string description), 1 H2S contain * Person

H2S has a fleet of pickup vehicles, which it uses to collect items from residents.

-> Vehicle(), 1 H2S contain * Vehicle

Repeat for every sentence of all N examples

Description: The LabTracker software helps (i) doctors...

Chain-of-thought

תוצרים עיקריים של המחקר

מסגרת עבודה

פיתוח Framework לביצוע מידול תחום בעזרת LLMs, המאפשר שימוש שיטתי במודלי שפה לצורך יצירת מודלים.

דאטה סט חדש

יצירת מאגר נתונים להערכת מידול תחום הכולל 10 תיאורים מילוליים (ברמה של קורסים אקדמיים) של מערכות יחד עם פתרונות שמומחי מידול יצרו כבסיס להשוואה.

השוואות מקיפות

ביצוע השוואה שיטתית בין 3 סגנונות פרומפטים ובין שלושה מודלים שונים (GPT-3.5 Davinci, GPT-3.5 Turbo, GPT-4).

תובנות מחקריות

זיהוי וניתוח טעויות נפוצות של LLMs בתהליך מידול תחום, המספקות בסיס לשיפורים עתידיים.

הערכת הפתרון ותהליך המחקר

דאטה סט

10 תיאורים מילוליים של מערכות, כאשר 2 שימשו כדוגמאות ו-8 לבדיקות

הערכה ידנית

השוואה בין המודלים שנוצרו על ידי LLMs לבין המודלים של המומחים נעשתה ידנית, כדי להתמודד עם המורכבות של הערכה סמנטית (כי ייתכנו כמה מודלים "נכונים").

שני בודקים מכותבי המאמר ערכו **שתי סבבי הערכה:**

- קודם כל הם בדקו זוג מודלים יחד, כל אחד באופן עצמאי.
- לאחר מכן השוו את תוצאותיהם, פתרו חילוקי דעות, והסכימו על **סט כללים להערכה אחידה**.
- רק אז הם עברו להעריך את שאר המודלים (כל אחד חצי מהם), ולאחר מכן **ביצעו תיקוף נוסף** לסבב השני.

מודלים

שימוש בשלושה מודלים: GPT-3.5 Davinci, GPT-3.5 Turbo, GPT-4

מערכת ניקוד

סיווג כל רכיב במודל לפי 4 קטגוריות:

קטגוריה	הסבר	ניקוד
c1	התאמה מלאה או סמנטית (מחלקה person לעומת מחלקה user)	1
c2	ייצוג שונה אך משמעות זהה (שימוש בboolean לעומת enum)	1
c3	נכונות חלקית/מושפע מטעות אחרת	0.5
c4	שגוי או חסר	0

כך התקבל **ניקוד משוקלל** עבור כל מודל - לכל אחת מהקטגוריות, ובעזרת ניקוד זה חושבו מדדי הביצוע הבאים:

Precision - כמה מהרכיבים שהמודל יצר היו נכונים (דיוק).

Recall - כמה מהרכיבים שציפינו להם הופיעו בפועל (שלמות).

F1 - ממוצע הרמוני שמאזן ביניהם.

תוצאות המחקר

עד כמה מודלי שפה גדולים מצליחים במידול תחום?

GPT-4 הראה ביצועים טובים יחסית:

- מחלקות - F1 של 0.76 (טוב מאוד)
- מאפיינים - F1 של 0.61 (בינוני)
- קשרים - F1 של 0.34 (חלש)

LLMs מצליחים לזהות ולבנות מחלקות בצורה טובה, ולעיתים גם מאפיינים, אך **מתקשים במיוחד בבניית קשרים בין רכיבים**. לעיתים הם גם מפספסים best practices.
יש ל-LLMs **דיוק (precision) גבוה**, כלומר כשהם כן מייצרים רכיב הוא לרוב נכון. אך הבעיה היא **Recall נמוך** - הם פשוט מפספסים הרבה דברים שצריך לכלול במודל. לכן, המודלים מבטיחים אבל רחוקים מאוטומציה מלאה.

איך Prompt Engineering משפיע על הביצועים?

- הוספת דוגמה (shot-1) שיפרה ביצועים משמעותית
- shot-2 לא נתן יתרון נוסף
- Chain-of-Thought דווקא **פגע בביצועים**

איזה מודל שפה מציג את הביצועים הטובים ביותר?

- GPT-4 הוביל בכל מדד ובכל סוג פרומפט
- ההבדלים היו בולטים במיוחד ב-Zero-shot, כלומר, GPT-4 מפגין הבנה כללית טובה יותר גם בלי דוגמאות

מסקנות וכיווני מחקר עתידיים

מסקנות המחקר

מודלי שפה גדולים כמו GPT-4 מפגינים הבנה טובה של הטקסט ויכולת לא רעה בזיהוי מחלקות ומאפיינים. עם זאת, הם אינם מצליחים לייצר מודל תחום שלם בעיקר בשל קשרים חלשים בין ישויות והתעלמות מעקרונות מידול מתקדמים כמו מחלקות אבסטרקטיות, דפוסי עיצוב נפוצים והפרדה בין מחלקות לאינומרציות.

הוספת צעדי חשיבה (CoT) לפרומפט לא שיפרה את ביצועי המודל, ואף פגעה, מאחר ש-CoT מתאים למשימות לוגיות בשלבים, בעוד שמידול תחום דורש הבנה של המערכת כולה.

הצעות להמשך מחקר

- פיתוח שיטות prompting טובות ומדויקות יותר שמתאימות ספציפית למשימות מידול
- בחירה אוטומטית של דוגמאות טובות לשלב בפרומפטים
- שילוב ידע מפורש על עקרונות מידול בתוך המודלים כדי שידעו לעקוב אחרי סטנדרטים מקצועיים ולהשתמש בדפוסי עיצוב נכונים
- כדאי להרחיב את הדאטה סט כך שיכלול תיאורי מערכות מגוונים יותר, כולל כאלה מהתעשייה, כדי לבדוק את הכלים בתנאים מציאותיים יותר

שאלות?