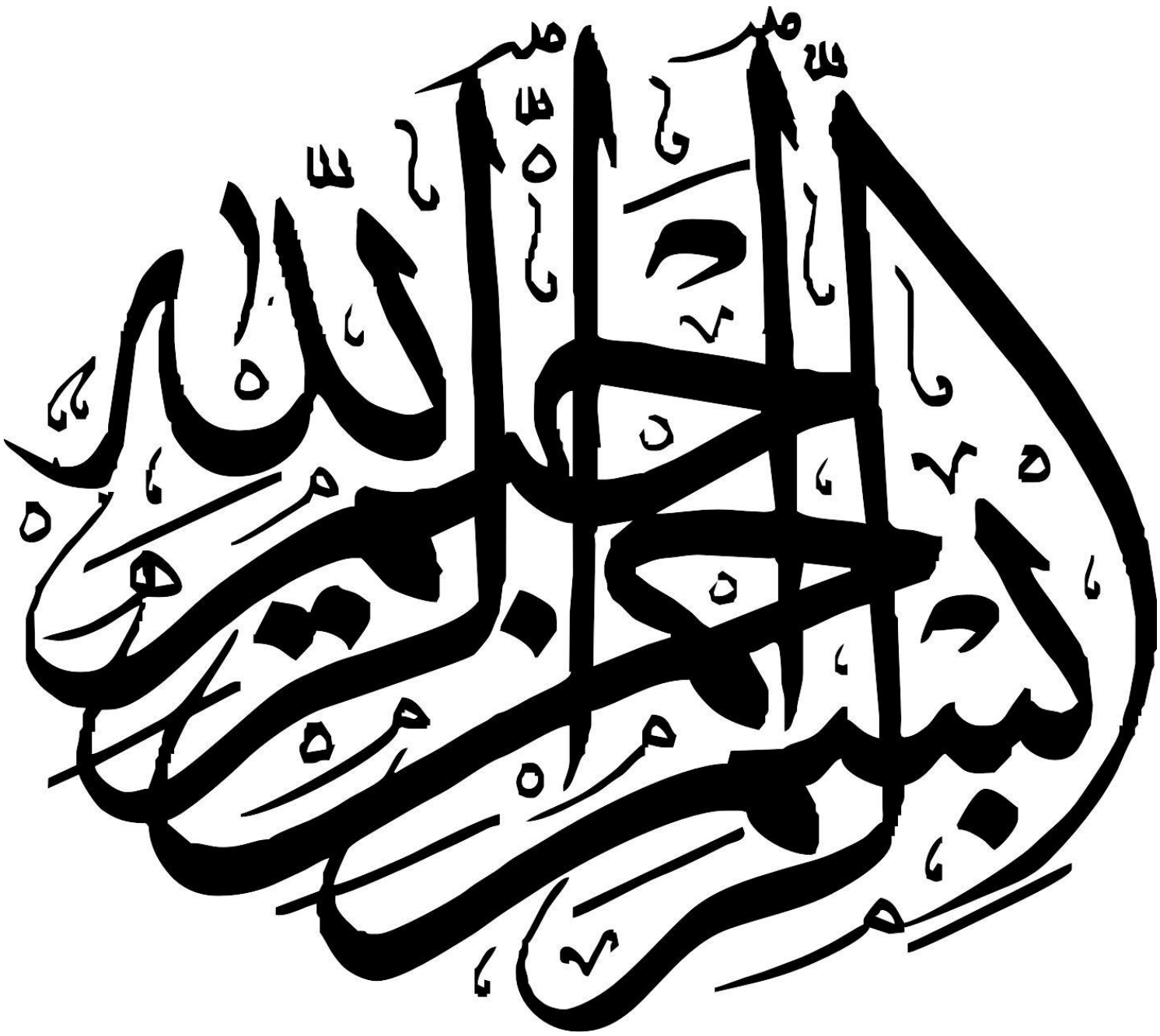

بسم الله الرحمن الرحيم

LLM Agents as Policy Makers (ABM)

Model Crafty

امیرحسین بیگدلی

زمستان 1404

فهرست مطالب

بخش اول

1.1. مقدمه

در حوزه مدل‌سازی سیستم‌های زمین(LSM) ، تعاملات میان انسان و محیط زیست همواره به صورت ساده‌انگارانه‌ای مدل شده است. در حالی که فرآیندهای فیزیکی (مانند چرخه کربن یا تغییرات دما) با دقت ریاضی بالایی شبیه‌سازی می‌شوند، "نهادهای سیاست‌گذار" که فرماندهان اصلی تغییر کاربری اراضی هستند، معمولاً به عنوان متغیرهای ثابت یا توابع خطی ساده در نظر گرفته می‌شوند. مسئله اصلی این است که نهادهای انسانی در دنیای واقعی، بر اساس داده‌های کامل عمل نمی‌کنند، دچار اینرسی (مقاومت در برابر تغییر) هستند و تصمیمات آن‌ها تحت تأثیر تفسیرهای متنی و تجربیات گذشته است.

2.1.چرا این مسئله اهمیت دارد ؟

عدم قطعیت در پیش‌بینی‌های اقلیمی اغلب ناشی از عدم قطعیت در رفتارهای انسانی است. اگر نتوانیم نحوه واکنش یک دولت به بحران کمبود مواد غذایی یا کاهش تنوع زیستی را به درستی مدل کنیم، سناریوهای انتشار آلاینده و مدیریت زمین ما با واقعیت فاصله زیادی خواهند داشت. استفاده از مدل‌های زبانی بزرگ (LLM)به ما این امکان را می‌دهد که برای اولین بار، "استدلال منطقی" و "تفسیر کیفی" را وارد مدل‌های عددی کنیم.

بخش دوم

2.1. ساختار ورودی‌ها(Inputs)

در سیستم طراحی شده، عامل هوشمند (Agent) اطلاعات را در سه سطح دریافت می‌کند:

1. **داده‌های عددی سیستم :**شامل میزان عرضه (Supply) و تقاضا (Demand) برای فرآورده‌های مختلف زمین (مانند غلات، گوشت و خدمات اکوسیستمی). این داده‌ها نشان‌دهنده شکاف (Gap) موجود در بازار هستند.
2. **بافتار تاریخی (Historical Context) :**مدل از طریق حافظه خود (Memory Buffer) می‌داند که در مراحل قبلی چه تصمیماتی گرفته و چه نتایجی حاصل شده است.
3. **اهداف و ماموریت‌ها(Goals) :** دستورالعمل‌هایی که به زبان طبیعی نوشته شده‌اند (مثلاً: "اولویت اول شما حفظ تولید گندم است در حالی که نباید مالیات از ۱۵٪ فراتر رود").

2.2. خروجی (Outputs)

خروجی مدل صرفاً یک عدد نیست، بلکه شامل دو بخش است:

- **بخش تبیینی (Reasoning) :**مدل توضیح می‌دهد که با توجه به افزایش تقاضای گوشت و کاهش مراتع، تصمیم گرفته است نرخ حمایتی را افزایش دهد.
- **بخش عملیاتی(Action) :** پارامترهای عددی (مانند نرخ Tax یاSubsidy) که مستقیماً به هسته مدل CRAFTY ارسال شده و نقشه کاربری زمین را در گام زمانی بعدی تغییر می‌دهد.

2.3. منبع و مشخصات داده ها

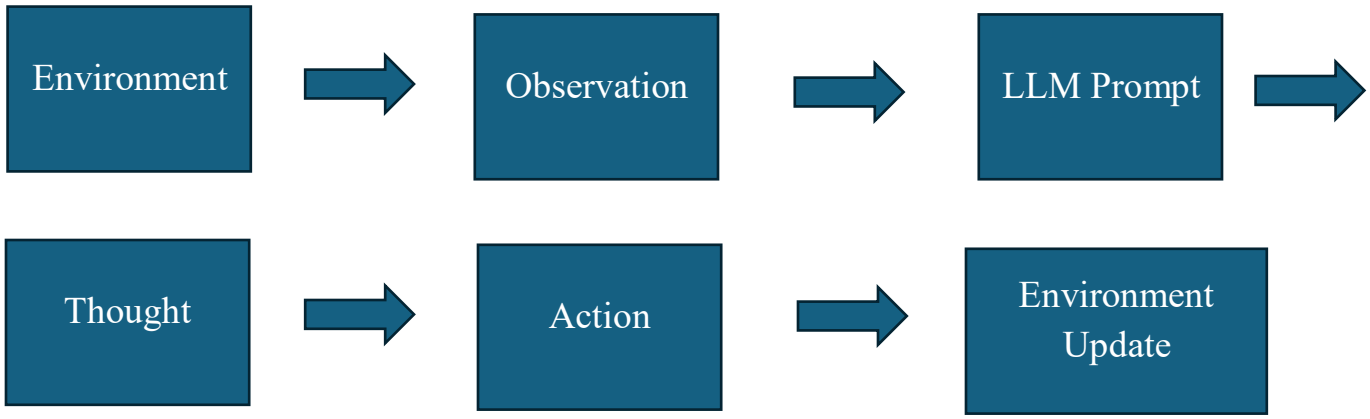
داده‌های اولیه از شبیه‌ساز CRAFTY استخراج شده‌اند که یک مدل مبتنی بر عامل (ABM) برای تغییر کاربری اراضی است. در کدهای پروژه (فایل DataCenter.java)، این داده‌ها جمع‌آوری و از طریق یک پل ارتباطی (Py4J) به پایتون منتقل می‌شوند تا توسط LLM پردازش شوند.

بخش سوم

3.1. معماری سیستم

روش پیشنهادی بر پایه معماری **ReAct (Reasoning + Acting)** بنا شده است. برخلاف مدل‌های کلاسیک که مستقیماً ورودی را به خروجی نگاشت می‌دهند، در اینجا یک لایه "تفکر" میانی وجود دارد.

شماتیک منطقی :



3.2. تحلیل کد و پیاده سازی

در فایل agents.py کلاسی به نام AgentExp مسئولیت مدیریت گفتگو با مدل زبانی را بر عهده دارد. از کلاس PromptTemplate در فایل templates.py برای تبدیل داده‌های خشک عددی به جملات قابل فهم برای هوش مصنوعی استفاده شده است.

مثال از فرآیند اجرایی : هنگامی که در فایل main.py تابع run_step() فراخوانی می‌شود، ابتدا وضعیت فعلی از جاوا خوانده شده، پرامپت ساخته می‌شود و خروجی LLM توسط تابع extract_tax_rate پارس می‌شود تا اطمینان حاصل شود که فرمت خروجی برای مدل جاوا معتبر است.

3.3. تغییرات داده شده نسبت به نسخه اصلی مقاله

در راستای افزایش کارایی، پایداری و قابلیت مانیتورینگ سیستم، تغییرات بنیادی در معماری ارتباطی و منطق اجرایی ریپازیتوری اصلی اعمال شد. این تغییرات شامل موارد زیر است:

1. **توسعه ماژول مدل های زبانی (Multy-LLM Support)**

در نسخه اصلی، محدودیت‌های زیادی در استفاده از مدل‌های مختلف وجود داشت. در نسخه فعلی:

- **پشتیبانی از Gemini :** با اصلاح فایل get_response.py و agents.py کلاس AgentExp، امکان فراخوانی APIهای Google Gemini فراهم شد.
- **معماری انعطاف‌پذیر :**سیستم به گونه‌ای تغییر یافت که بتوان تنها با تغییر یک پارامتر در کانفیگ، بین مدل‌هایGPT ، Gemini یا مدل‌های محلی (Local) جابجا شد، بدون اینکه منطق اصلی شبیه‌سازی تغییر کند.

2. **پیاده‌سازی سیستم لاگر اختصاصی (CSV Logging System)**

برای تحلیل دقیق‌تر رفتار عامل‌ها و امکان بازتولید نتایج (Reproducibility):

- یک سیستم **Logger اختصاصی** طراحی شد که تمام تعاملات شامل «پرامپت ارسالی»، استدلال عامل (Reasoning) و «تصمیم نهایی» را به همراه برچسب زمانی در یک فایل **CSV** ذخیره می‌کند.
- این قابلیت اجازه می‌دهد تا پس از اتمام شبیه‌سازی، بتوان نمودارهای سری زمانی از نحوه تغییر تفکر عامل در طول سال‌های مختلف را به راحتی استخراج کرد.

3. **مدیریت متمرکز پیکربندی و محیط (Environment Control)**

برای جلوگیری از پراکندگی تنظیمات در فایل‌های مختلف:

- کنترلر Config: فایل config.properties در سمت جاوا و سیستم مدیریت env در سمت پایتون (فایل key.py و connection_handler.py) یکپارچه شدند.

4. **استاندارد سازی نصب و راه اندازی (Automation Scripts)**

به منظور تسهیل در اجرای پروژه در محیطهای مختلف:

- **اسکریپت‌های اتوماسیون :**فایل‌های اجرایی run_project.sh (برای لینوکس/مک) و run_project.bat (برای ویندوز) اضافه شدند. این اسکریپت‌ها به طور خودکار ابتدا سرور جاوا را راه‌اندازی کرده و سپس کلاینت پایتون را متصل می‌کنند، که خطاهای انسانی در ترتیب اجرای کامپوننت‌ها را حذف می‌کند.
- **فایلrequirements.txt :** تمام نیازمندی‌های پایتون (از جمله py4j برای ارتباط با جاوا و google-generativeai برایGemini) به صورت استاندارد لیست شد.

بخش چهارم

4.1. تحلیل نتایج

نشان می‌دهد که عامل‌های مبتنی بر LLM تمایل دارند "احتیاط" به خرج دهند. برخلاف الگوریتم‌های بهینه‌سازی که ممکن است تغییرات ناگهانی و شدیدی در نرخ‌ها ایجاد کنند، LLMها مانند مدیران انسانی ترجیح می‌دهند تغییرات را به صورت تدریجی (مثلاً ۲ درصد در هر مرحله) اعمال کنند. این امر منجر به پایداری بیشتر در سیستم‌های اجتماعی-اقتصادی می‌شود.

4.2. محدودیت ها

- **توهم (Hallucination):** گاهی مدل اعدادی خارج از محدوده منطقی پیشنهاد می‌دهد که نیاز به لایه‌های کنترلی (Validation Layers) در کد دارد.
- **پنجره حافظه:** با طولانی شدن شبیه‌سازی، حجم تاریخچه تصمیمات زیاد شده و ممکن است مدل جزئیات مراحل اولیه را فراموش کند.
- **هزینه و زمان:** فراخوانی API برای هر گام زمانی شبیه‌سازی در مقیاس بزرگ، زمان‌بر است.

4.3. ایده‌های توسعه و کاربرد عملیاتی

- **استفاده در سامانه‌های دولتی:** این پروژه می‌تواند به عنوان یک "جعبه شن (Sandbox) "برای سیاست‌گذاران عمل کند تا قبل از تصویب یک لایحه، بازخوردهای احتمالی سیستم زمین را مشاهده کنند.

- **عامل‌های چندگانه (Multi-Agent):** توسعه سیستم به گونه‌ای که چندین عامل LLM (مثلاً نماینده بخش کشاورزی و نماینده محیط زیست) با هم مذاکره کنند تا به یک نرخ مالیات مشترک برسند. این کار در فایل‌های LLMInstitution.java و AgriInstitution.java به عنوان زیربنا آماده شده است.