

CognitiveRAG

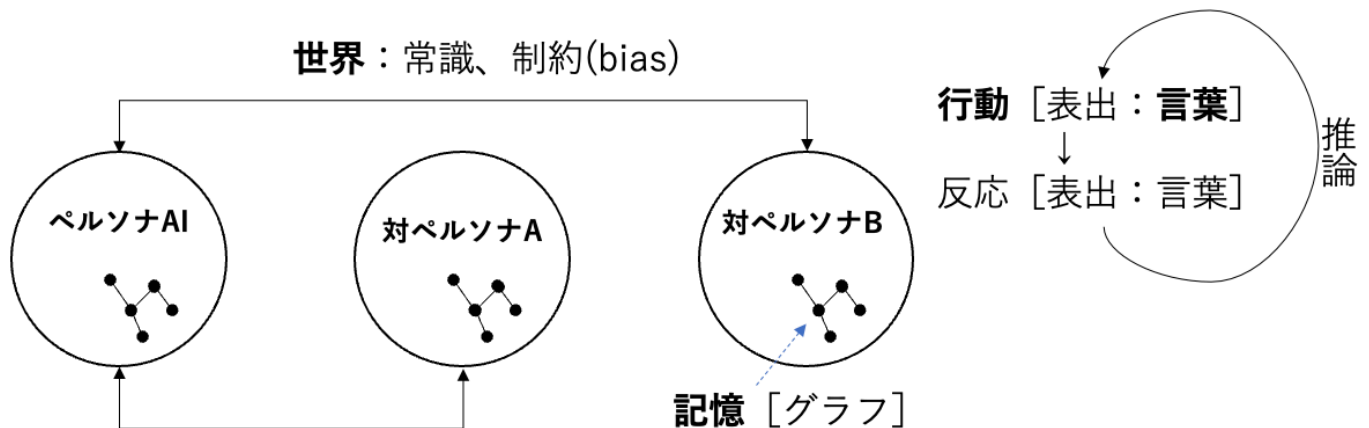
[メモ] 2025-10-11 matsuda@symbolics.jp

備考：以下のアイデアは松田の案を中心にChatGPT5にも大幅に支援していただき構想中のもの。

デザインプロセス

- ・ グラフ構造をデータベースとするRAGデザインを構想 [松田]
- ・ 知識の**記憶**としての置き換え [松田]
- ・ 記憶にまつわる問題—想起、連想、更新 [by ChatGPT5]
- ・ Python上で様々なモデルを検証⇒コードが複雑化して、版管理困難になる
- ・ 別件で「介護向けRAG」の案件が入る。
- ・ いったん、休止していたCognitiveRAGを再度引っ張り出す。ただし、。。。
- ・ Mathematica上に再実装を試みる
 - 現在、記憶の想起、連想（ワープ）モデルをグラフ上に構築・実験中
 - 記憶（グラフ）が動的に変更されるモデルを想定。
 - 連想をサブグラフ間のジャンプで代替。

CognitiveRAG



注：記憶（データベース）＆メタパラメータはエージェントごとに異なる

プロンプト

因果律について。追記。

本モデルの前提には「行動」と「その結果」、さらに加えて「次の結果」につながる、action連鎖が念頭にありました。確認。

・ 単語モデルベース（ここでいま考えているものと同じでなくともよい）でaction連鎖のモデルを作ることとは可能ですか。

・ ただし単語選択が次の単語選択に至る中に「世界」（場）を挟みたい。ここで世界は一種の常識ある

は制約と考えてよく、それ自身がグラフでもたんなる表でも構わない。

お願い。

・この内容にそった疑似コードを15行程度で作ってみて。 ・モデル名に名前をつけて。

```
# LWAC: LexiWorld Action Chain
Input: start_word w0, start_state s0, world W (graph/table of
constraints/affordances), horizon T
1  w ← w0; s ← s0; seq ← [w]
2  for t = 1..T:
3      A ← Affordances(w, s, W)                # 世界wが許す行為/語候補（制約でフィルタ）
4      Scored ← { a : Score(a | w, s, W) for a in A } # 言語遷移×世界制約の合成スコア
5      a* ← argmax_a Scored[a]                  # 次の「行為としての語」選択
6      (s', r) ← WorldTransition(s, a*, W)      # 世界が更新され、結果r（中間的事象）が生じ
る
7      w' ← Realize(r, W)                       # 事象rを語彙へ写像（叙述/言語化）
8      seq.append(w'); w ← w'; s ← s'          # 連鎖を更新
9      if StopCriterion(w, s, W): break         # 目標達成/矛盾/エネルギー切れなど
10 return seq

# 参照的定義（実装時の指針）
11 Affordances(w,s,W):= {a ∈ V | ConstraintsSatisfied(a,s,W)}
12 Score(a|w,s,W):= λ·LMprob(a|w) + μ·WorldCompat(a,s,W) + ν·PlanValue(a,s,W)
13 WorldTransition(s,a,W):= ( Dynamics_W(s,a), Result_W(s,a) )
14 Realize(r,W):= ArgMax_v Similarity( SemEmbed(v), EventEmbed(r,W) )
15 StopCriterion(w,s,W):= GoalReached(s) or NoMove(A) or t ≥ T
```

Daniel C. Dennett の “robot R1” [フレーム問題]

行為語

```
["receive-mission",
 "detect-low-battery",
 "locate-battery-room",
 "enter-room",
 "identify-battery-on-wagon",
 "ignore-side-effects",
 "grasp-wagon-handle",
 "pull-wagon-out",
 "battery-exits-room",
 "bomb-also-exits-room",
 "no-reasoning-about-bomb",
 "move-to-safe-area",
 "bomb-detonates-outside",
 "mission-failure"]
```

AIの支援を得たプロセスデザインについて

- **Mathematica**を使う。
- **短いプロンプト**：CognitiveRAGに関する「膨大なやりとり」が前提（ChatGPT5はチャット間およびチャット内でコンテキストを記憶している）

- **短いコード**：ロジックに傾注できる。
- デバッグモニタの作成が容易。（ChatGPT5でエラーなし一発で作成）
- 可視化が容易。（ChatGPT5でエラーなし一発で作成）