

字符模式 SAC v4.0.0 决策摘要与双迭代方案

- 作者: GaoZheng
- 日期: 2025-09-27
- 版本: v1.0.0

摘要

在 v3.0.1“前/后缀词包可命中”的基础上，引入“分段级词包双向演化”：先做压缩迭代，用“摘要词包”逐段吸收“正文词包”；再做扩展迭代，从高密度摘要反向重建“正文词包”，并同步更新更高层摘要。目标是把“字符级稀疏奖励”结构化为“段级词包事件流”，实现“可审计压缩 → 可审计重建 → 文法风格补全”的长上下文生成，上线重点关注吞吐、SLA 与合规可回放。

- What: 在 v3.0.1 的基础上，引入分段级词包双向演化：
 - 压缩迭代：用“摘要词包”逐段吸收“正文词包”，得到高密度结构摘要 S_k ；
 - 扩展迭代：从高密度摘要 S_k 反向展开出一批“候选正文词包” B_k^* ，并同步更新更高层摘要 S_{k+1}^* 。
- Why: 把“字符级稀疏奖励”进一步结构化为“段级词包事件流”，实现“可审计压缩 → 可审计重建 → 文法风格补全”的长上下文生成；上线关注吞吐、SLA、合规可回放三线同向。

1. 核心对象与算子（可计算语义）

1.1 词包对象（分段级）

- 正文词包 B_i : 来自第 i 个正文分段（或段块）的词/短语集合；
- 摘要词包 S_k : 由前 k 段（或指定上下文）压缩得到的高权词/短语集合；
- 用户提问词包 B_q : 把“本次问题”抽取为词包，参与末端压缩与引导展开。

词包由 AC/Trie + 规范化映射（别名/全半角/大小写/同义）获得，命中语义承袭 v3.0.1（顺序敏感的非交换短语支持）。

1.2 半环耦合 (带权 KAT)

- 取 Log-Viterbi 半环 $S = (\mathbb{R} \cup \{-\infty\}, \oplus = \max, \otimes = +)$;
- 词包权: $\log w(\omega) = \log \mu(\omega) + \log \text{idf}(\omega) + \mathbf{1}[\text{sim} > \tau] \cdot 0 + \text{pack_w}$;
- tests (合规硬闸) : 不过闸 $\Rightarrow -\infty$; 通过 $\Rightarrow 0$.

1.3 两个主算子 (段级 Option 宏动作)

- 压缩算子 \mathcal{C} (聚合/去冗) :

$$S_{k+1} = \text{TopM}(S_k \otimes \alpha \oplus B_{k+1} \otimes \beta), \quad \alpha, \beta > 0,$$

其中 TopM 按 \oplus 选择前 M 个高权词包, 内含去重/别名合并/冷却衰减。

- 扩展算子 \mathcal{E} (检索/重建) :

$$(B_{k+1}^*, S_{k+2}^*) = \mathcal{E}(S_{k+1}) = \left(\text{Retrieve/Decode}(S_{k+1}), \mathcal{C}(S_{k+1}, B_{k+1}^*) \right).$$

`Retrieve/Decode` 可优先检索 EKB (内存库), 不足再小模型解码 (受 tests 与掩码约束)。

两算子均按 Option (宏动作) 实现: 一外步触发, 内部最多 K_{\max} 子步, 终止=命中或封顶; 计费与奖励按外步折算, 日志保留子步轨迹。

2. 双迭代流程 (压缩 \rightarrow 问题对齐 \rightarrow 扩展 \rightarrow 文法补全)

2.1 压缩迭代 (正文 0...n)

- 初始化: $S_0 = \mathcal{C}(\emptyset, B_0)$;
- 递推: $S_i = \mathcal{C}(S_{i-1}, B_i)$, $i = 1, \dots, n$;
- 问题对齐: $S_{n+1} = \mathcal{C}(S_n, B_q)$.

2.2 扩展迭代 (自顶向下重建)

- 展开 1: $(B_{n+1}^*, S_{n+2}^*) = \mathcal{E}(S_{n+1})$;
- 展开 j : $(B_{n+j}^*, S_{n+1+j}^*) = \mathcal{E}(S_{n+j})$, $j = 2, \dots, n$;
- 汇总正文重建: $\mathcal{B}^* = \{B_{n+1}^*, \dots, B_{2n}^*\}$.

2.3 文法风格补全 (线性化 → 风格器)

1. 线性化：按段序与依赖把 B^* 组装为候选段落；
 2. 风格器：小模型（或规则）执行连词/指代/时态/标点补全；
 3. 长上下文输出：得到段落化的长上下文 LLM 应答草稿；
 4. tests：事实/敏感/配额合规硬闻；失败回退到更保守的检索版段落。
-

3. 伪代码（端到端）

```
# 压缩阶段
S = compress(empty, B_0, α, β, M)
for i in 1..n:
    S = compress(S, B_i, α, β, M)
S = compress(S, B_q, α_q, β_q, M)    # 问题对齐，权重可不同

# 扩展阶段 (Option 宏动作, 外步)
B_star = []
for j in 1..n:
    (B_new, S_next) = expand_option(S, K_max, EKB, small_decoder, tests)
    B_star.append(B_new)
    S = compress(S, B_new, α_r, β_r, M)    # 同步更新更高层摘要

# 文法风格补全
Draft = linearize(B_star, order="temporal->causal->definition")
Answer = style_completion(Draft, constraints={tense, coref, punctuation})
return Answer
```

4. 训练/奖励 (SAC × 事件流)

- 外步奖励 r_t :

$$r_t = \underbrace{语义}_{S_t} + \underbrace{词法}_{δ_t} - \underbrace{成本}_{C_t},$$

其中 δ_t 来自词包命中 (Log-Viterbi 权) 并受语义门控 τ 、IDF/二字降权、冷却约束；长度成本含 L_h, L_p, K_{\max} ，守住吞吐。

- MDP 闭合：压缩/扩展均为 Option；策略更新在外步进行；子步轨迹仅入日志。
- MDQ (可学参数) : $\tau, \alpha, \beta, M, K_{\max}$ 及 pack_w 缩放；更新

$$\Delta = Q(\partial \mathcal{J} / \partial \alpha) - \lambda_{\text{comm}} \sum |[G_i, G_j]| \pi_j.$$

5. 配置 (新增/扩展)

```
{  
  "packs": {  
    "body_packs_path": "data/body_packs.json",  
    "summary_packs_path": "data/summary_seed.json",  
    "normalize": { "alias": true, "casifold": true, "fullwidth": true }  
  },  
  "compression": { "alpha": 0.7, "beta": 1.0, "topM": 64, "cooldown_window": 32 },  
  "question_align": { "alpha_q": 0.9, "beta_q": 1.2 },  
  "expansion": {  
    "enabled": true, "k_max": 3, "hit_mode": "union", "alpha_r": 0.6, "beta_r": 1.0  
  },  
  "attn": { "L_h": [2,3,4,6], "L_p": [3,4,5], "cost": { "lambda": 0.1 } },  
  "reward": { "tau_semantic_gate": 0.75, "lambda_lex": 0.30, "downweight_bigram": 0.6 }  
}
```

6. 日志与回放 (可审计)

- 压缩: cmp_step, topM, merged_aliases, pack_ids_added, pack_ids_dropped
- 扩展: opt_id, option_duration, ekb_hits, decoder_hits, suffix_topo_hit, pack_id, hit_len
- 质量/成本: S_t, delta_t, L_h, L_p, K_max, len_cost
- 安全: tests_passed, blacklist_hits, rollback_reason

7. 复杂度与性能 (可上线)

- AC/Trie 匹配：均摊近 $\mathcal{O}(1)$ ；检索优先级：packs → catalog → decoder；
 - $k_{\max} \leq L_p$ 保证扩展成本受控；TopM 控制摘要体积；
 - 建议把词包匹配与检索放在独立 worker/C 扩展，启用状态缓存；CPU 占比目标 < 10%。
-

8. KPI 与 A/B (验收强口径)

- 词法合规：`word_noncompliance` $\downarrow \geq 30\%$ ；
 - 覆盖：术语/短语覆盖 +8–15pp；
 - 稳定：收敛步数 $\downarrow \geq 15\%$ 、训练方差 $\downarrow \geq 20\%$ ；
 - 产线：P95/QPS 达标；回放/回滚 100%；
 - 压缩–扩展一致性：BLEU/ROUGE between $\cup B_i$ 与 $\cup B_j^*$ \uparrow ，风格器后事实 tests 通过。
-

9. 风险与 Runbook

风险	根因	处置
过压缩丢关键	TopM 过小、IDF 过强	提高 TopM/降低 IDF 缩放； <code>pack_w</code> 上调关键域
扩展幻觉	decoder 过早介入	先检索 EKB，再小模型；tests 未过直接弃用
刷分投机	高频短语/同一 pack 重复	开启 cooldown、二字降权、提升 τ
吞吐下滑	k_{\max} 或 L_p 过大	绑定 $k_{\max} \leq L_p$ ，并限制 L_p 分布
词包污染	低质/敏感条目	上线前静态审计 + 黑名单 + kill_switch

10. 兼容与回滚

- 关扩展：`expansion.enabled=false` 即退回 v3.0.1；
- 退回单词：无 `summary_packs` 或失败时，压缩退化为 Catalog-only；
- 统一双缓冲 + 金丝雀 + KPI 守门，异常自动回滚；ledger 记录版本与差异。

许可声明 (License)

Copyright (C) 2025 GaoZheng

本文档采用[知识共享-署名-非商业性使用-禁止演绎 4.0 国际许可协议 \(CC BY-NC-ND 4.0\)](#)进行许可。