字符模式 SAC 的工程实现与数学化描述v1.0.0

- 作者：GaoZheng

- 日期：2025-09-26
- 版本：v1.0.0

摘要

版本 v1.0.0 聚焦最小可用字符级 SAC：定义观测/动作/奖励与回放结构，给出策略与双 Q 网络的参数化与损失，提供训练循环与指标记录的标准模板。强调能跑通、易复现与可度量，为后续版本的稳态与性能优化打下基线。

- 本文系统化描述了在“字符模式”下的软演员-评论家（Soft Actor-Critic, SAC）实现，用数学语言刻画观测构造、策略与价值网络、奖励构成以及与中文词典相结合的拓扑逻辑（raw_action 与 bigram 的前向/后向拓扑）。
- 核心贡献包括：将策略输入统一为 prev + ⟨sep⟩ + chapter 的字符级观测、引入历史左扩机制保证源串前缀命中、以 data/word_length_sets.json 提供的词长并集驱动可变长度的“后缀命中”判定，并将其纳入奖励与日志注记。

关键词：SAC；字符级强化学习；中文词法；可变长度后缀；历史左扩；Top-p 采样

1 引言
字符级文本决策常因观测粒度小、词法边界难以对齐而产生奖励稀疏与学习不稳。本文提出的工程化方案在字符模式下引入两类拓扑：
（1）前缀左扩，保证源串 source 的头两个字符在词表中命中；
（2）基于词长并集 U 的“后缀命中”拓扑，驱动 raw_action 与 bigram 的语义停表（hit-stop）。配合 SAC 的熵正则框架与 Top-p 策略采样，有效提升训练信号与对齐稳定性。

2 问题设定与环境
2.1 观测空间与编码
记时刻 t 的观测为 $O_t = (s_t, \chi_t, i_t)$。在字符模式下：

- s_t 为上一轮摘要的预览（仅取末尾 ≤ 1 字用于构造 source 与可视化）。
- χ_t 为“当前目标字符”（从教师序列中取，非空）。
- i_t 为步索引。

输入编码由 CharTokenizer 生成：

$$x_t = [\texttt{<bos>}] \ \oplus \ \mathrm{clip}(s_t) \ \oplus \ [\texttt{<sep>}] \ \oplus \ \chi_t \ \oplus \ [\texttt{<eos>}]$$

并裁剪至 `max_observation_length`。

伪代码：

```
function ENCODE_OBSERVATION(prev_summary_char, chapter_char):
    tokens = [<bos>] + encode(prev_summary_char)
    tokens += [<sep>] + encode(chapter_char) + [<eos>]
    return clip(tokens, max_len)
```

2.2 动作与解析

策略一次生成 action token 序列 y_t，经解析器 OperationParser 还原为结构化动作序列 $A_t = (a_1, \ldots, a_{m_t})$。字符模式下我们仅取 canonical_summary 的首字作为预测字符（参与 bigram 评价），结构化操作不作用于环境状态（仅在章节模式生效）。

2.3 环境转移与缓存

给定 A_t 后，环境进行资本与预算更新：

$$C_t = \Gamma(C_{t-1}, A_t), \qquad B_t = B_{t-1} - \sum_{a \in A_t} c(a)$$

同时记录 $\mathrm{metrics}_t$，并产出下一观测 O_{t+1} 与奖励 r_t，加入重放缓存 $\mathcal{D}$。

3 中文词法约束与拓扑

3.1 词表与长度集合

- 词表 Catalog = {data/chinese_name_frequency_word.json, data/chinese_frequency_word.json}，均为只读。
- 允许后缀长度并集 U 从 data/word_length_sets.json 读取：$U = \mathrm{union.lengths}$（例如 $U = \{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13\}$）。

3.2 源串与前缀左扩

字符模式下源串定义为：

$$\mathrm{source}_t = \mathrm{prev}_t \oplus \chi_t$$

若 $|\mathrm{source}_t| \geq 2$ 且 $\mathrm{source}_t[:2] \notin \mathrm{Catalog}$，则执行“历史左扩”（至多 N 次）：

```
function EXTEND_PREV_FOR_PREFIX_HIT(prev, chapter, history_pairs, N):
    source = prev + chapter
    if len(source) < 2: return prev
    step = 0
    while (source[:2] not in Catalog) and (step < N) and history_pairs.has_left(step):
        pair = history_pairs.left(step)          # pair = head+tail, tail 与 prev 的首字对齐
        if prev and not prev.startswith(pair.tail):
            step += 1; continue
        prev = pair.head + prev
        source = prev + chapter
        step += 1
    return prev
```

其中 $N = \texttt{character\_history\_extension\_limit} = 16$，来自 `res/config.json`。

3.3 raw_action 的后缀拓扑（去重首字、命中即停）
设策略首字预测为 c，`future_chars` 为后续真值字符流。定义序列 q 的生成：

```
function EXTEND_RAW_ACTION_SUFFIX(c, future_chars, U):
    q = dedup_head_repeat(c)         # 若未来序列以 c 重复则去重一次
    for ch in future_chars:
        q += ch
        for L in sort_desc(U ∩ {1..len(q)}):
            seg = tail(q, L)
            if is_cjk(seg) and in_catalog(seg):
                return q, seg, annotate(seg)  # 命中即停
    # 未命中时保留最长连续 CJK 片段作为回退
    return q, longest_cjk_tail(q, max(U)), annotate_optional()
```

注：重复情形如“辑辑…”将去重为“辑…”，若“脉络”在 U 中命中，则到“脉络”为止停止拓扑。

3.4 bigram 的前向拓扑（继承 raw_action）
定义 bigram 候选串 $s = \chi_t \oplus q$，以相同的 U 做后缀命中：

```
function FORWARD_EXTEND_BIGRAM(chapter, q, future_chars, U):
    s = chapter + q
    best2 = tail(s, min(2, len(s)))
    for ch in future_chars:
        s += ch
        for L in sort_desc(U ∩ {1..len(s)}):
            seg = tail(s, L)
            if is_cjk(seg) and in_catalog(seg):
                return seg, s        # 返回用于奖励与日志的后缀命中
    return best2, s                  # 回退为最后两字
```

bigram 注记以“后缀命中”为核心，如 data/chinese_frequency_word.json#236703。

4 奖励函数与评分方案

4.1 质量/词法/洁净度分量

设 $\mathrm{metrics}_t$ 为当前摘要质量指标：similarity、coverage_ratio、novelty_ratio、lexical_cosine、lexical_js_similarity、garbled_ratio、word_noncompliance_ratio 等。令非线性放大算子 $\mathcal{N}_\gamma(x) = 1 - (1 - x)^\gamma$，则：

$$
\begin{aligned}
Q_t &= 0.6\,\mathcal{N}_{4.0}(\texttt{similarity}) + 0.3\,\mathcal{N}_{4.0}(\texttt{coverage\_ratio}) + 0.1\,\mathcal{N}_{4.0}(\max(0, \texttt{novelty\_ratio})),\\
L_t &= 0.15\,\mathcal{N}_{3.5}(\texttt{lexical\_cosine}) + 0.10\,\mathcal{N}_{3.5}(\texttt{lexical\_js\_similarity}),\\
P_t &= 0.5\,\mathcal{N}_{5.0}(\texttt{garbled\_ratio}) + 0.7\,\mathcal{N}_{5.0}(\texttt{word\_noncompliance\_ratio}),\\
S_t &= Q_t + L_t - P_t.
\end{aligned}
$$

4.2 字符模式加成与 bigram 奖励

令 $\chi_t^{\mathrm{soft}} = \max(0, Q_t + L_t)$。bigram 奖励基于 U 的后缀命中：

$$
\delta_t = \begin{cases}
1.0, & \exists L \in U,\ \mathrm{tail}(s, L) \in \mathrm{Catalog},\\
0.5, & \mathrm{tail}(s, 1) = \mathrm{target\_char},\\
0, & \text{otherwise}.
\end{cases}
$$

字符模式下对基础/潜在分量加成：

$$
B_t^{\mathrm{char}} = B_t + 0.5\,\chi_t^{\mathrm{soft}} + \delta_t, \qquad \Delta_t^{\mathrm{char}} = \Delta_t + 0.25\,\chi_t^{\mathrm{soft}}.
$$

4.3 总奖励

令成本罚项 λ_t 与预算罚项 ψ_t：

$$
c_t = \sum_{a \in A_t} c(a), \quad B_t = B_{t-1} - c_t, \qquad \lambda_t = \omega_c\,(\bar{c}_t\ (\text{终局})\ \text{else}\ c_t), \quad \psi_t = \beta\ \max(0, -B_t).
$$

综合：

$$
R_t = V(C_t) - \lambda_t - \psi_t + S_t + \mathbf{1}_{\mathrm{char}} \cdot \left(\Delta_t + 0.5\,\chi_t^{\mathrm{soft}} + 0.25\,\chi_t^{\mathrm{soft}} + \delta_t\right).
$$

5 策略/价值网络与 SAC 优化

5.1 策略网络 π_θ

输入 x_t 经过嵌入与编码 GRU 得到 h^{enc}，再经解码 GRU 自回归产生 y_t：

$$y_t \sim \pi_\theta(\cdot \mid x_t), \qquad y_t = (y_{t,1}, \dots, y_{t,m}).$$

对 logits 施加合规 mask 与温度 τ_c，首步分布经 Top-p（$p = 0.98$）筛选候选，记录概率与 log-prob。

5.2 双 Q 网络 Q_ϕ, Q_ψ

状态与动作分别嵌入，经均值掩码得到 u, v 并拼接，经 MLP 输出标量 Q 值。目标网络采用指数滑动平均：

$$\theta' \leftarrow \tau\,\theta + (1-\tau)\,\theta'.$$

5.3 损失函数

目标期望：

$$y = r + \gamma(1-d)\,\mathbb{E}_{a'\sim\pi}\big[\min(Q_{\phi'}(s',a'), Q_{\psi'}(s',a')) - \alpha\,\log\pi(a' \mid s')\big].$$

评论家损失：

$$\mathcal{L}_Q = \mathrm{MSE}(Q_\phi(s,a), y) + \mathrm{MSE}(Q_\psi(s,a), y).$$

策略损失与温度自适应：

$$\mathcal{L}_\pi = \mathbb{E}_{s\sim\mathcal{D}} \sum_i w_i\big(\alpha\,\log p_i - Q_\phi(s,a_i)\big), \qquad \mathcal{L}_\alpha = -\log\alpha \cdot (H_{\text{tgt}} - H_{\text{emp}}), \quad H_{\text{tgt}} = \kappa\,\log|\mathcal{A}(s)|.$$

6 日志与注记规范

- source 行：显示 $\text{source} = \text{prev} + \text{chapter}$，必要时附“前缀最长命中”注记（词表/编号）。
- raw_action 行：显示长度与文本，并附“后缀命中”注记（命中词 + 词表/编号）。
- bigram 行：以 $\text{chapter} \oplus \text{raw\_action}$ 为基，显示“后缀命中”。
  注记示例： (后缀"喃喃": data/chinese_frequency_word.json未命中) 或 data/chinese_frequency_word.json#236703。

7 复杂度与实现细节

- 历史左扩：最坏 $O(N)$，$N = \texttt{character\_history\_extension\_limit}$（默认 16）。
- 后缀命中：每步遍历 U 的降序长度，$O(|U|)$。U 来自 data/word_length_sets.json（建议 $|U| \le 12$ 以控时）。
- 词典装载：缓存与热重载（监控 mtime），避免频繁 IO。
- 只读保证：运行时不回写 data/sample_article_lexical.json。

8 复现实务

- 配置：res/config.json 与 config_template.json 中的 $\texttt{character\_history\_extension\_limit} = 16$。

- 词长并集：data/word_length_sets.json 的 `union.lengths`。
- 代码入口：src/train_demo.py —— ArticleEnvironment, DemoSACAgent, TextPolicyNetwork, TextQNetwork 等。
- 文档一致性：脚本 scripts/check_docs_sync.py 用于变更提示（可选）。

9 结论

本文给出字符级 SAC 的一套可复用工程化方案，用“历史左扩 + 可变长度后缀命中”将中文词法知识注入到奖励与日志注记中，稳定训练并提升可解释性。今后可在子词/词级混合粒度、动态 U 自适应与更细致的对齐约束上扩展。

附录 A 记号表

- O_t：观测；x_t：Token 序列；y_t：动作 Token 序列；A_t：解析后动作；C_t：认知资本；B_t：预算；V/P：资本价值/潜力；
- U：允许后缀长度集合；Catalog：词表并集；δ_t：bigram 奖励；χ_t^{soft}：质量软信号；
- prev/χ_t/source：上一摘要预览/目标字符/源串；q/s：raw_action 序列/ bigram 串。
