

Bottom-Up 建模方法论

AI 大模型 Token 消耗预测 · 详细实战指南

维度	数值 / 说明
文档定位	Bottom-Up 预测法的完整操作手册
目标读者	AI 产品经理 / 成本架构师 / 数据科学家
配套 PPT	AI_Cost_Value_Analysis.pptx (Part 02, P.13)
配套摘要	Executive_Summary.pdf
核心公式	$T = \Sigma (\text{任务类型} \times \text{单次 token} \times \text{调用次数}) \times \text{放大系数}$
核心方法	七步法 + 双校验 + 历史回归
文档版本	v1.0 · 2026.09

阅读指南

维度	数值 / 说明
第 1 页	本文档的定位、核心公式与七步法概览
第 2 页	客服场景完整拆解 (从单任务到季度聚合)
第 3 页	编程助手场景 + Agent / RAG 放大系数详解
第 4 页	校准方法 + 常见陷阱 + Top-Down 双校验
第 5 页	速查表 + Excel 模板字段说明

一、Bottom-Up 方法定位与核心公式

定义：Bottom-Up（自下而上）是一种从微观任务出发，逐层聚合到宏观总量的预测方法。它把 Token 总需求拆解为具体的任务清单，再分别量化每个任务的单次消耗与调用频次，最终加总得到预测值。

适用场景：产品迭代期的成本分摊、A/B 测试、精细化运营、定价模型、工程团队容量规划。

核心公式：

$$T_{\text{period}} = \sum_i [N_{\text{users}} \times \text{freq}_i \times (L_{\text{in}_i} + L_{\text{out}_i}) \times \text{amp}_i]$$

Where:

T_{period} = total Token consumption in the period

N_{users} = active users / sessions of task i

freq_i = call frequency per user / session

L_{in_i} = input tokens per call (system + user + context)

L_{out_i} = output tokens per call

amp_i = Agent / RAG / multimodal amplification (1.0 - 10x)

与 Top-Down 的对比

维度	Top-Down 自上而下	Bottom-Up 自下而上
起点	业务 KPI (DAU / GMV / 转化)	任务清单 (Task Inventory)
聚合方向	宏观 → 微观	微观 → 宏观
公式	$T = \text{DAU} \times \text{频次} \times \text{长度}$	$T = \sum \text{任务}_i \times \text{单次 token} \times \text{次数}$
数据需求	产品大盘数据	单次调用日志、用户行为日志
精度	中 (战略级)	高 (运营级)
适用周期	季度 / 年度规划	周 / 月迭代
失败模式	遗漏 Agent 放大效应	高估早期渗透率

核心原则：两种方法必须独立预测并相互校验。差异 > 30% 时，需重新审视假设。

二、七步法 + 客服场景完整拆解

七步法概览

Step	动作	关键产出
1	业务场景 → 任务清单	Task Inventory (10-30 个原子任务)
2	单次任务 → Token 拆解	System / User / Context / Output 长度表
3	调用频次估算	日均 / 周均 / 月均 × 用户分层
4	基础聚合公式	$T_base = \Sigma (\text{用户} \times \text{频次} \times \text{单次 token})$
5	Agent / RAG 放大	amp 系数 (1.0 - 10x)
6	场景敏感性分析	P10 / P50 / P90 三档
7	历史校准 + 回归	回归方程 + 残差诊断

案例：智能客服场景完整拆解

场景描述：某电商客服系统日均 50,000 次会话，AI 处理 70%。

Step 1-2: 任务清单 + 单次 Token 拆解

任务 ID	任务名称	System	User	Context	Output	单次合计
T01	问候 / 路由	300	20	200	50	570
T02	FAQ 问答	500	80	300	120	1,000
T03	订单查询	600	50	1,500	100	2,250
T04	退换货流程	800	120	2,000	300	3,220
T05	投诉处理	1,000	200	3,500	500	5,200
T06	多轮 Agent (T04→T05)	1,500	400	5,000	800	7,700
加权平均	—	—	—	—	—	≈ 3,200

数据来源：基于开源客服场景基准 + 行业平均值估算。实际部署请用真实日志采样。

Step 3: 调用频次估算

用户分层	DAU	日均会话/人	占比	加权单次 token
高频 (VIP)	5,000	3.5	10%	3,800
中频 (普通)	25,000	1.2	50%	3,000
低频 (偶发)	30,000	0.3	40%	2,500
AI 承接率	70%	—	—	—
加权平均	—	1.0	100%	≈ 3,200

三、聚合公式 + 放大系数

Step 4: 基础聚合 (单日)

T_day = SUM (DAU_segment x freq_segment x tokens_per_call x ai_share)

T_day = 5,000 x 3.5 x 3,800 + 25,000 x 1.2 x 3,000 + 30,000 x 0.3 x 2,500

= 66.5M + 90.0M + 22.5M

= 179.0M tokens / day

x 70% AI share = 125.3M AI tokens / day

T_month = 125.3M x 30 = 3.76B tokens / month

Step 5: Agent / RAG 放大系数

客服场景涉及多轮 Agent 与 RAG 文档检索，需要乘以放大系数：

组件	放大机制	典型 amp	本案例取值
基础对话	单轮问答	1.0x	1.0x
多轮对话	历史 + 重述	1.5-2.5x	1.8x
RAG 检索	Top-K 文档拼入 context	2.0-4.0x	2.5x
Agent 工具调用	多次 LLM 调用 + 工具结果	3.0-8.0x	4.0x
多模态	图像转 token	1.5-3.0x	—
综合系数 (本案例)	RAG + 多轮 Agent	—	amp = 2.5x

Corrected monthly consumption:

T_month_corrected = 3.76B x 2.5 = **9.40B tokens / month**

Step 6: 场景敏感性 (P10 / P50 / P90)

场景	假设变化	amp 系数	月消耗
乐观 P10	Agent 渗透率 60%, 多模态启用	3.5x	18.4B
基准 P50	当前规划假设	2.5x	9.4B
悲观 P90	Amp 仅含 RAG, 单轮为主	1.5x	5.6B

启示：基准 / 乐观差距 ≈ 2x，必须为容量规划预留 buffer。

■ 四、编程助手案例 + 校准方法

案例 2：AI 编程助手场景

场景：某 500 人研发团队部署 Copilot 类工具，月活 400 人。

任务	触发频次 / 人·天	单次 In	单次 Out	放大系数	日均 token
代码补全 (Tab)	30	500	80	1.0	17,400
对话式问答	5	1,500	600	1.5	15,750
代码重构 Agent	1	2,500	2,000	3.0 (Agent)	13,500
代码审查	0.5	3,000	1,500	2.0 (RAG)	4,500
Bug 定位多轮	0.3	2,000	1,200	4.0 (Agent)	3,840
日均合计 / 人	—	—	—	—	≈ 55,000

$T_{\text{month}} = 400 \text{ users} \times 55,000 \times 22 \text{ workdays} = 484\text{B tokens / month}$

编程场景的特征

调用频次极高 (30+ 次/日)，单次 token 较小
Agent / RAG 集中在 20% 的任务，但贡献 60% 的 token
杠杆点：上下文缓存命中率 → 直接降低 In token 30-50%

Step 7: 历史校准 + 回归方法

核心思想：用过去 4-12 周的真实日志，建立回归方程：

$\log(T_{\text{actual}}) = b_0 + b_1 \cdot \log(\text{DAU}) + b_2 \cdot \log(\text{avg_len}) + b_3 \cdot \text{Agent_share} + e$

校准动作：

- ① 收集 4 周历史数据，按周聚合
- ② 拟合 OLS 回归，记录 R² 与残差
- ③ 用预测周的实际值与模型预测值对比，MAPE < 20% 为合格
- ④ 不合格时回到 Step 1-3 重检任务清单与 amp 假设

常见陷阱 (Top 5)

陷阱	症状	规避方法
遗漏 Agent 放大	预测值系统性偏低 30-50%	强制 amp ≥ 2.0x 假设
上下文重复计费	同一文档被多次检索	引入 KV cache 命中率
频次高估	按会话峰值而非均值	用 7 天滑动平均
任务遗漏	新功能上线后无对应任务	季度任务清单 review
单 token 漂移	模型升级后长度变化	版本变更后重测基准

五、速查表与 Excel 模板字段

Bottom-Up 预测速查表 (7 行)

步骤	输入	公式 / 动作	输出
1. 任务清单	业务场景	拆解为 10-30 个原子任务	Task ID + 名称
2. Token 拆解	代表性 prompt	采样 100-500 次取 P50	In / Out / 合计
3. 频次估算	用户行为日志	按分层加权平均	freq_segment
4. 基础聚合	1+2+3	$\Sigma \text{用户} \times \text{频次} \times \text{token}$	T_base
5. 放大系数	场景特征	查表 / 经验估算	$\text{amp} \in [1.0, 10]$
6. 敏感性	P10/P50/P90	调整 amp 与频次	三档预测
7. 回归校准	历史日志	OLS 回归 + MAPE 验证	回归方程

Excel 模板字段说明

字段	类型	示例	说明
task_id	String	T01	任务唯一 ID
task_name	String	FAQ 问答	任务中文名
user_segment	Enum	高频/中频/低频	用户分层
dau_segment	Integer	5000	该分层日活
freq_per_user	Float	3.5	单用户日均会话
tokens_in_p50	Integer	500	输入 token 中位数
tokens_out_p50	Integer	120	输出 token 中位数
amp_factor	Float	2.5	Agent/RAG 放大系数
ai_share	Float	0.7	AI 承接率
daily_tokens	Formula	$=B * C * (E + F) * H * I$	日均 token
monthly_tokens	Formula	$=J * 22 \text{ 或 } 30$	月消耗
p10_p50_p90	Float	1.5/2.5/3.5	敏感性三档

三步实施清单 (今日可启动)

- 第 1 天：召集产品 / 工程 / 数据三方，对齐业务场景清单 (1-2 小时)
- 第 2-3 天：抽样 100 次代表性 prompt，测 In/Out token 中位数
- 第 4-5 天：拉取 DAU + 频次数据，按分层加权平均
- 第 6 天：基础聚合 + 放大系数估算，输出 P10/P50/P90
- 第 7 天：与 Top-Down 预测对比，差异 > 30% 时触发假设 review

总投入：约 1 人 × 1 周；产出：可落地的季度 Token 预测 + 容量规划依据

—— END ——