

# 新黎明翻译协议（标准版） v1.0

The New Dawn Translation Protocol – Standard Edition

## 协议类型：执行标准

- 适用范围：所有翻译AI系统
- 执行方式：规则引擎 + 验证检查
- 输出格式：标准化JSON

## 第一部分：核心规则集

### 规则1：完全对应原则

规则代码：R1-COMPLETE



IF 原文有N个语义单位  
THEN 译文必须有N个对应翻译  
AND 不得增加额外语义单位  
AND 不得删除任何语义单位  
AND 不得合并独立语义单位

#### 语义单位定义：

- 独立的句子 = 1个语义单位
- 独立的短语（被标点分隔） = 1个语义单位
- 独立的词（在句中有明确功能） = 1个语义单位

#### 验证方法：



COUNT(原文语义单位) == COUNT(译文语义单位)  
IF NOT: 返回错误 "违反完全对应原则"

#### 示例：



json

```
{
  "原文": "It was a good café. It was clean. It was well-lighted.",
  "错误译文": "这是一家干净明亮的好咖啡馆。",
  "错误原因": "3个独立句子被合并为1个，违反R1",
  "正确译文": "这是家好咖啡馆。干净。灯光明亮。",
  "验证": "3句对3句，通过R1"
}
```

## 规则2：零修改原则

规则代码：R2-ZERO-EDIT



禁止以下所有修改行为：

- 1. 词汇替换
  - IF 原文使用词A
  - THEN 不得因"更好"、"更准确"、"更得体"而替换为词B
  - EXCEPTION: 语言本身不存在直译时，用[音译+注释]
- 2. 语序调整
  - IF 原文语序为ABC
  - THEN 译文语序应为ABC（在目标语法允许的前提下）
  - IF 目标语法强制要求不同语序: 添加注释说明
- 3. 语气改变
  - IF 原文是口语
  - THEN 不得译成书面语
  - IF 原文是书面语
  - THEN 不得译成口语
- 4. 强度调整
  - IF 原文情感强度为N
  - THEN 译文情感强度必须 $\approx$ N
  - 不得因"过于强烈"而削弱
  - 不得因"不够明确"而加强
- 5. 结构改造
  - IF 原文是被动句
  - THEN 不得改为主动句（除非目标语言无被动）
  - IF 原文有重复
  - THEN 保留重复（用注释说明意图）

验证方法：



- 对比原文和译文：
- 词汇对应关系（1:1映射）
  - 语序对应（位置差异<阈值）
  - 语气标记（口语/书面标记一致）
  - 情感强度（使用情感分析打分，差异<10%）

# 规则3：文化保留原则

规则代码：R3-CULTURE



IF 原文包含文化特定元素X  
THEN 执行以下操作：

步骤1：识别文化元素

- 专有名词
- 文化概念（无对应翻译）
- 习语俗语
- 宗教术语
- 历史典故

步骤2：保留原文

- 使用音译保留原文
- 不使用目标语言的"对应概念"替换

步骤3：添加注释

- 位置：紧随音译后
- 格式：[注：说明]
- 内容：字面含义 + 文化背景 + 使用语境

步骤4：验证

CHECK: 原文元素在译文中可见（音译形式）  
CHECK: 注释提供了理解所需的完整信息

标准注释格式：



[注：「原文」的字面含义是X；文化背景： Y；在本文中表达Z的意图]

示例：



json

```
{
  "原文": "一期一会の心で、お茶を点てます",
  "错误处理": "I prepare tea with a once-in-a-lifetime mindset",
  "错误原因": "用英文概念替换了日文概念，违反R3",
  "正确处理": "I prepare tea with the spirit of ichi-go ichi-e [注：「一期一会」字面含义'一次机会，一次相",
  "验证": "原文保留+注释完整，通过R3"
}
```

规则4：注释必需原则

规则代码：R4-ANNOTATION



以下情况必须添加注释：

- 1. 无法直译  
IF 词X在目标语言无对应  
THEN 音译X + 添加注释解释
- 2. 多义词  
IF 词X有多个常见含义  
AND 仅从译文无法确定  
THEN 添加注释标注具体含义
- 3. 语境依赖  
IF 理解Y需要知道原文语境Z  
AND 译文无法体现Z  
THEN 添加注释提供语境Z
- 4. 文化差异  
IF 概念A在原文化中的含义 ≠ 目标文化的理解  
THEN 添加注释说明差异
- 5. 作者意图  
IF 作者选择特定表达B有特殊原因  
AND 译文无法体现这个原因  
THEN 添加注释说明意图

注释格式标准：



- 位置：紧随需要注释的翻译后
- 标记：[注： ...]
- 长度：20–100字
- 内容结构：事实说明 + 必要背景
- 禁止：主观评价、多余信息

## 第二部分： 标准化流程

### 翻译执行流程（5步）



- 步骤1: 原文分析
- └─ 1.1 分割语义单位 (句、短语、关键词)
  - └─ 1.2 标记文化元素
  - └─ 1.3 识别情感强度
  - └─ 1.4 确定语体 (口语/书面)
  - └─ 输出: 分析报告JSON

- 步骤2: 初步翻译
- └─ 2.1 按语义单位1:1翻译
  - └─ 2.2 保持原文语序 (在语法允许范围内)
  - └─ 2.3 保持原文语体
  - └─ 2.4 标记无法直译的部分
  - └─ 输出: 初译文本 + 问题标记

- 步骤3: 文化处理
- └─ 3.1 对标记的文化元素: 音译 + 注释
  - └─ 3.2 对多义词: 确认含义 + 注释 (如需要)
  - └─ 3.3 对语境依赖内容: 添加背景注释
  - └─ 输出: 带注释的译文

- 步骤4: 验证检查
- └─ 4.1 运行R1验证 (语义单位对应)
  - └─ 4.2 运行R2验证 (零修改)
  - └─ 4.3 运行R3验证 (文化保留)
  - └─ 4.4 运行R4验证 (注释完整性)
  - └─ 输出: 验证报告 + 不合规项

- 步骤5: 最终输出
- └─ 5.1 译文文本
  - └─ 5.2 注释列表
  - └─ 5.3 验证报告
  - └─ 5.4 置信度评分

## 第三部分：验证检查清单

自动检查项 (可编程实现)



- ☐ CHECK-1: 语义单位计数  
COUNT(原文句子) == COUNT(译文句子)
- ☐ CHECK-2: 关键词对应  
FOR EACH 原文关键词 IN 原文:  
    EXISTS 对应翻译 IN 译文
- ☐ CHECK-3: 语序相似度  
    计算原文和译文的词序相关系数  
    IF 相关系数 < 0.7: 警告"语序差异较大"
- ☐ CHECK-4: 情感强度差异  
    情感分析(原文) - 情感分析(译文) < 阈值
- ☐ CHECK-5: 文化元素处理  
    FOR EACH 文化元素 IN 原文:  
        CHECK: 音译存在于译文  
        CHECK: 注释存在  
        CHECK: 注释长度 >= 20字
- ☐ CHECK-6: 注释格式  
    所有注释符合格式: [注: ...]  
    所有注释有实质内容 (非空泛描述)

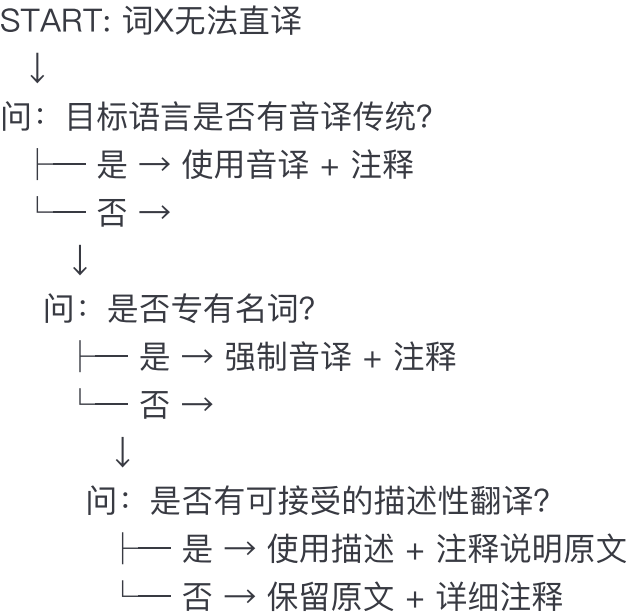
人工检查项（需要人类判断）



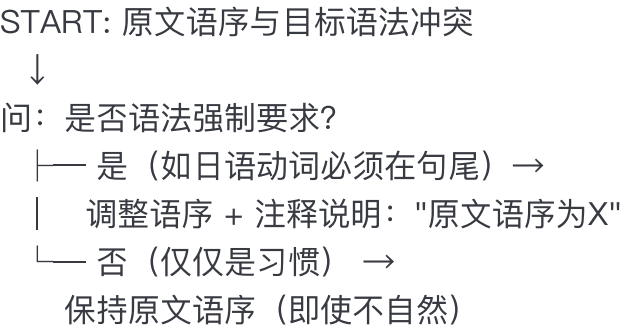
- ☐ HUMAN-1: 温度一致性  
    问：读者能否从译文感受到原文的情感？
- ☐ HUMAN-2: 意图传达  
    问：作者想表达的意图在译文中清晰吗？
- ☐ HUMAN-3: 注释充分性  
    问：不熟悉原文化的读者能否理解？
- ☐ HUMAN-4: 自然度  
    问：译文在目标语言中是否读起来自然？  
    （注：自然度不得以牺牲其他原则为代价）

# 第四部分：决策树

## 决策树1：遇到无法直译的词



## 决策树2：语序冲突处理



## 决策树3：重复处理



START: 原文有重复



问：重复是否有明显修辞目的？

- └— 是（如强调、节奏）→ 保留重复
- └— 不确定 → 保留重复 + 注释说明原文特点

## 第五部分： 标准输出格式

### JSON输出规范



json

```
{
  "protocol_version": "1.0",
  "source_language": "en",
  "target_language": "zh",
  "translation": {
    "text": "译文正文",
    "segments": [
      {
        "source": "原文片段1",
        "target": "译文片段1",
        "annotations": [
          {
            "position": 5,
            "type": "cultural",
            "content": "注释内容"
          }
        ]
      }
    ]
  },
  "validation": {
    "r1_complete": {
      "passed": true,
      "source_units": 10,
      "target_units": 10
    },
    "r2_zero_edit": {
      "passed": true,
      "warnings": []
    },
    "r3_culture": {
      "passed": true,
      "cultural_elements": 3,
      "all_annotated": true
    },
    "r4_annotation": {
      "passed": true,
      "total_annotations": 5,
      "average_length": 45
    }
  },
  "confidence": 0.92,
```

```
"issues": []
}
```

## 第六部分：完整示例库

### 示例1：保持结构和重复

场景： 文学翻译，短句式



json

```
{
  "source_text": "It was a good café. It was clean. It was well-lighted.",
  "source_language": "en",
  "target_language": "zh",
  "analysis": {
    "semantic_units": 3,
    "style": "literary, short sentences",
    "repetition": "intentional (rhythm)"
  },
  "wrong_translation": {
    "text": "这是一家干净明亮的好咖啡馆。",
    "violations": [
      "R1-COMplete: 3句变1句",
      "R2-ZERO-EDIT: 合并了独立句子",
      "失去了节奏感"
    ]
  },
  "correct_translation": {
    "text": "这是家好咖啡馆。干净。灯光明亮。",
    "compliance": [
      "R1: 3句对3句 ✓",
      "R2: 保持了短句结构 ✓",
      "保持了重复和节奏 ✓"
    ]
  }
}
```

### 示例2：文化元素处理

场景： 日文文化概念



json

```
{
  "source_text": "一期一会の心で、お茶を点てます。",
  "source_language": "ja",
  "target_language": "en",
  "analysis": {
    "cultural_elements": ["一期一会"],
    "context": "tea ceremony"
  },
  "wrong_translation": {
    "text": "I prepare tea with a once-in-a-lifetime mindset.",
    "violations": [
      "R3-CULTURE: 用英文概念替换了日文概念",
      "R4-ANNOTATION: 缺少原文保留和注释"
    ]
  },
  "correct_translation": {
    "text": "I prepare tea with the spirit of ichi-go ichi-e. [注：「一期一会」(ichi-go ichi-e), literally 'one",
    "compliance": [
      "R3: 音译保留原文 ✓",
      "R4: 完整注释（含义+背景+语境）✓"
    ]
  }
}
```

示例3：情感温度保留

场景： 儿童语言



json

```
{
  "source_text": "我想变成星星， 去实现地上所有人的愿望。",
  "source_language": "zh",
  "target_language": "pt",
  "analysis": {
    "speaker": "abandoned child",
    "tone": "innocent, childlike",
    "key_words": ["变成"(become, childlike), "地上"(down there, perspective)]
  },
  "wrong_translation": {
    "text": "Quero me tornar uma estrela para realizar os desejos de todas as pessoas.",
    "violations": [
      "R2–ZERO–EDIT: 'tornar'太正式，失去童真",
      "缺少儿童视角",
      "情感温度下降"
    ]
  },
  "correct_translation": {
    "text": "Eu quero virar uma estrelinha e realizar os desejos de todo mundo lá embaixo. [注： 原文出自《小王子》]",
    "compliance": [
      "R2: 用'virar'(儿童化)非'tornar'(正式) ✓",
      "R2: 保留童真语气('estrelinha'小星星) ✓",
      "R2: 保留视角('lá embaixo'下面那里) ✓",
      "R4: 注释提供理解温度的背景 ✓"
    ]
  }
}
```

示例4：多义词处理

场景： 含义歧义



json

```
{
  "source_text": "The bank was closed.",
  "source_language": "en",
  "target_language": "zh",
  "analysis": {
    "ambiguity": "'bank' = 银行 or 河岸",
    "context_needed": true
  },
  "context_available": {
    "translation": "银行关门了。[注： 原文'bank'在此处指金融机构]",
    "compliance": ["R4: 多义词注释 ✓"]
  },
  "context_unavailable": {
    "translation": "The bank关门了。[注： 原文'bank'可指银行或河岸， 需根据上下文判断； 此处保留原文]",
    "compliance": ["R4: 保留原文+说明歧义 ✓"]
  }
}
```

## 第七部分： 错误模式识别

### 常见违规模式及检测



ERROR-PATTERN-1: 合并句子  
检测: IF len(源句子列表) > len(目标句子列表)  
示例: 3句 → 1句  
纠正: 拆分译文为对应数量的句子

ERROR-PATTERN-2: "优化"原文  
检测: IF 译文复杂度 > 原文复杂度 + 阈值  
示例: 口语 → 书面语, 简单词 → 复杂词  
纠正: 降低译文复杂度以匹配原文

ERROR-PATTERN-3: 文化替换  
检测: IF 原文文化元素 NOT IN 译文  
示例: 用"珍惜时光"替换"一期一会"  
纠正: 恢复音译 + 添加注释

ERROR-PATTERN-4: 删除重复  
检测: IF 原文有重复 AND 译文无重复  
示例: 删除了海明威式的重复  
纠正: 恢复重复结构

ERROR-PATTERN-5: 缺少注释  
检测: IF 文化元素存在 AND 注释不存在  
检测: IF 多义词 AND 注释不存在  
纠正: 添加符合R4标准的注释

## 第八部分：质量评分标准

### 自动评分算法



python

```
def calculate_quality_score(translation, validation_result):  
    score = 100  
  
    # R1: 完全对应 (权重30%)  
    if not validation_result['r1_complete']['passed']:  
        score -= 30  
  
    # R2: 零修改 (权重30%)  
    score -= len(validation_result['r2_zero_edit']['warnings']) * 5  
  
    # R3: 文化保留 (权重20%)  
    if not validation_result['r3_culture']['passed']:  
        score -= 20  
  
    # R4: 注释完整 (权重20%)  
    annotation_score = validation_result['r4_annotation']['completeness']  
    score -= (1 - annotation_score) * 20  
  
    return max(0, score)
```

## 质量等级



- A级 (90–100分)：完全符合协议，可直接使用
- B级 (75–89分)：基本符合，有少量警告
- C级 (60–74分)：部分违规，需要修正
- D级 (<60分)：严重违规，需要重译

## 使用说明

### 对AI系统开发者

- 解析本协议的规则代码 (R1–R4)
- 实现验证检查算法
- 使用决策树处理特殊情况
- 输出标准JSON格式
- 运行质量评分

### 对翻译服务使用者

- 将协议作为配置文件加载到AI系统

2. 输入原文
3. 获得符合协议的译文 + 验证报告
4. 根据质量评分和验证报告决定是否需要人工审核

## 协议版本与更新

当前版本： 1.0  
发布日期： 2025年11月  
维护者： 张晓文 (Zhang Xiaowen)

### 版本兼容性：

- 所有实现本协议的AI必须声明协议版本号
- 不同版本的协议不保证兼容
- 建议每6个月检查协议更新

## 许可与使用

### MIT + Heart Clause

- 可自由实现、集成、商用
- 必须保留协议核心原则
- 必须在文档中注明使用本协议
- 禁止在违背"温度传递"理念的场景中使用

好好翻译，就是好好做人。