

2025 年度红山开源系列创新大赛
红山开源大模型创意应用竞赛
参赛作品策划书

作品名称:	“净语” 智能敏感词检测系统
参赛单位:	武警工程大学
队长姓名:	姜佳泽
团队成员:	王赛、程关潼、胡玉强、张宇泽
联系方式:	19720292818
填报日期:	2025 年 11 月 7 日

目 录

一、背景.....	1
二、产品设计思路.....	1
三、技术方案.....	2
（一）整体架构.....	2
（二）技术栈选型.....	4
（三）核心检测算法	4
1.规则匹配引擎的算法架构	5
2.大模型引擎的算法架构	6
（四）数据流与安全性	6
四、功能使用描述.....	7
（一）数据输入.....	7
（二）数据处理.....	7
（三）结果生成.....	8
（四）系统维护.....	8
五、应用价值.....	9
（一）从“表层规则匹配”到“深层语义理解”的突破	9
（二）解决“效率与精度”难以兼顾的难题	9
（三）高度的灵活性与扩展性	9
（四）快速部署能力与广泛应用前景	10
六、创新点.....	10
（一）双引擎驱动，兼顾精准与高效	10

（二）容器化部署，跨环境一键启动	11
（三）模块化设计，支持功能拓展与灵活接入	11
（四）双轨检测策略，灵活适配多场景	11
（五）多格式输入，一站式文档解析	12
（六）敏感词库管理，实时动态更新	12
七、核心团队人员介绍	12
八、核心成果展示	13

一、背景

在信息内容快速增长、广泛传播的时代，敏感词检测被广泛应用于内容审核、评论过滤、社区管理、舆情监控等场景，在保障平台运营、净化社交环境、维护社会稳定等方面发挥着重要作用。然而，传统的敏感词检测技术严重依赖关键词匹配与敏感词库，在面对日益复杂的语义安全威胁时，其检测效果呈现显著衰减。具体而言，传统方法难以有效应对谐音变体（如“法溪斯”）、拼音替代（如“法 xi 斯”）及隐晦表达（需结合上下文判断）等策略，其原因在于传统的规则匹配仅停留在字符表层，而无法理解背后的深层语义。因此，发展能够深度融合语义理解的新一代智能敏感词检测系统，已成为应对当前内容安全严峻挑战的迫切需求。

为破解上述技术瓶颈，团队开发了“净语”智能平台信息敏感词检测系统，创新采用“规则匹配快速初筛+大语言模型智能检测”的技术范式，兼顾检测速度、准确率和召回率，构建了一个融合高效规则引擎与大语言模型深度语义分析的系统性解决方案。

二、产品设计思路

本系统在架构设计上采用“模块化+容器化”设计思想，开发 Web 界面作为前端，使用 FastAPI 作为后端，使用 Ollama 作为大模型运行平台，这种架构设计有利于快速部署和二次开发。

在实际场景中，我们认为大多数待审查内容应属于正常内容，敏感内容应只占小部分。所以在检测逻辑上，采取了“双重检测，智能协同”的设计思想，由规则匹配引擎快速过滤并“放行”正常内容，而后由大模型引擎智能检测存疑内容，二者协同实现了在保障高吞吐性能的同时，显著

提升复杂场景下的识别准确率，解决传统检测工具在效率与精度之间的失衡问题。

具体而言，系统主要分为两个层面：

在用户交互层面，设计了简洁的 web 前端界面（如图 2-1 所示），集成实时文本输入、多格式文档（TXT/PDF/DOCX/OCR）上传、敏感词库管理、“高效-严格”双模式选择（默认为高效模式）等功能，检测结果通过可视化模块实时呈现。所有功能均提供 API 接口，用户可根据需要灵活调用。

在核心架构层面，系统采用“AC 自动机与 DFA 协同检测”、“规则匹配与大语言模型协同检测”的双协同机制，用户提交内容后，首先经由“AC 自动机+DFA”规则引擎进行毫秒级初筛；对于初筛中发现的存疑内容，系统将自动触发 LLM 引擎进行深度语义推理与上下文关联分析。此举有效弥补了单一规则匹配在应对谐音、变体及语义伪装时的固有缺陷。此外，独立的词库管理模块使管理员能够实时更新敏感词规则，使系统具备持续的更新与适应能力。

三、技术方案

“净语”系统的技术方案旨在构建一个协同、高效与可扩展的智能检测平台，其实现路径基于分层与解耦的架构思想，深度融合了现代 Web 开发框架与前沿人工智能技术。

（一）整体架构

在整体技术架构与选型层面，系统采用前后端分离的设计模式。前端基于 HTML/CSS/JS 构建响应式用户界面，后端通过 FastAPI 将文本检测、

文档检测、敏感词库管理等功能封装为不同 API 接口。用户不仅可以通过前端界面直接进行操作，还可以根据需要将系统功能灵活集成至其他平台。后端将大语言模型（Qwen2.5:7B INT4）通过 Ollama 框架进行本地化服务化封装，为敏感词检测提供深层语义理解，并确保了数据处理过程的内网闭环与隐私安全。整体架构如图 3-1：



图 3-1 整体架构图

（二）技术栈选型

技术栈选用及具体技术说明如表 3-1:

表 3-1 技术栈表，对系统各分段使用的技术栈进行介绍和说明

类别	技术/工具	说明
后端技术	FastAPI	现代化的 Python Web 框架，支持异步处理和自动生成 API 文档
	Uvicorn	基于 uvloop 的高性能 ASGI 服务器，用于运行 FastAPI 应用
	Pydantic	用于数据验证、序列化和设置管理的库，提升类型安全性和开发效率
	PyPDF2	用于解析和操作 PDF 文档
	python-docx	用于读取和解析 DOCX 格式文档
	antiword	用于解析旧版 DOC 文档的命令行工具
	pytesseract	Python 封装的 OCR 库，用于调用 Tesseract 进行图像文字识别
	Tesseract OCR	开源的图片文字识别引擎，支持多语言识别
	AC 自动机	多模式字符串匹配算法，适用于高效敏感词或关键词检索
	DFA（确定性有限自动机）	用于实现高效的文本过滤与状态机匹配，常用于内容审核或关键词检测
	文本预处理	包括字符归一化、全半角转换、大小写统一、变体统一等，提升文本匹配准确性
前端技术	HTML5	用于构建语义化的网页结构
	CSS3	实现现代化、响应式的页面样式设计
	JavaScript (ES6+)	编写客户端交互逻辑，支持模块化和现代语法
	Fetch API	用于在浏览器中发起异步 HTTP 请求，与后端 API 通信
	Drag & Drop API	实现文件拖拽上传功能，提升用户体验
AI 技术	Ollama	轻量级本地大语言模型运行环境，便于部署和管理 LLM
	Qwen2.5:7b	通义千问 2.5 版本，70 亿参数的量化模型，支持本地高效推理
	Prompt Engineering	通过优化提示词设计，提升大模型输出质量与任务准确性
部署技术	Docker	容器化技术，实现应用及其依赖的隔离与可移植性
	Docker Compose	用于定义和运行多容器 Docker 应用（如后端、前端、数据库等）
	WSL（Windows Subsystem for Linux）	在 Windows 上运行 Linux 环境，便于开发和部署基于 Linux 的应用

（三）核心检测算法

在核心检测算法的协同设计上，采取“规则快速过滤、LLM 智能识别”的协同判别机制，实现了规则匹配与语义理解的深度融合。

1.规则匹配引擎的算法架构

作为系统的第一道防线，规则匹配引擎承担了毫秒级初筛与高吞吐过滤的重任。为兼顾匹配效率与规则灵活性，系统采用了“AC 自动机 + DFA”的双引擎融合架构。其中，AC 自动机凭借其基于 Trie 树与失败指针的机制，能够高效完成大规模敏感词库的多模式精确匹配，时间复杂度接近 $O(n)$ 。而 DFA（确定有限状态机）则通过其状态转移模型，负责处理复杂规则、变体表达及字符插入等非标准文本的识别。

规则匹配引擎的检测流程是：首先对输入文本进行预处理，而后由 AC 自动机对输入内容进行快速检测，若发现敏感词则直接输出；若未发现敏感词，则交由 DFA 对可能存在的变体模式进行二次检测。二者的协同工作，构成了系统应对模式化敏感内容的高效屏障。其总体架构如图 3-2。

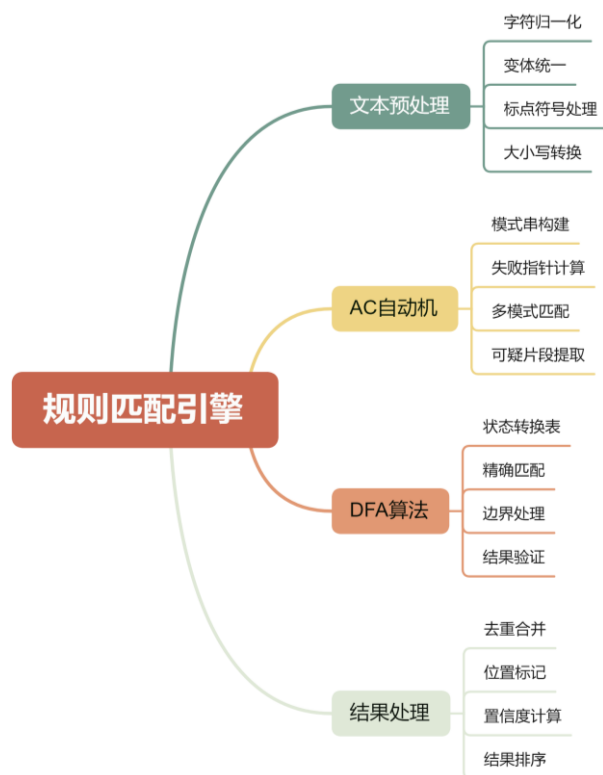


图 3-2 规则匹配引擎架构图

2.大模型引擎的算法架构

对于规则匹配引擎初筛后的存疑内容，系统调度本地部署的大语言模型进行深度语义解析。系统选用 Ollama 作为本地化模型运行框架，其“模型即服务”的特性与 API 接口，为系统提供了数据安全、低延迟且可私有化部署的智能大脑。通过精心设计的提示工程，LLM 能够精准执行上下文推理与风险意图识别，有效弥补了规则引擎在应对语义伪装时的固有缺陷。经过综合评估，系统选用 Qwen2.5:7B INT4 混合量化模型，在检测精度与推理速度之间实现了最优平衡。其总体架构如图 3-3。

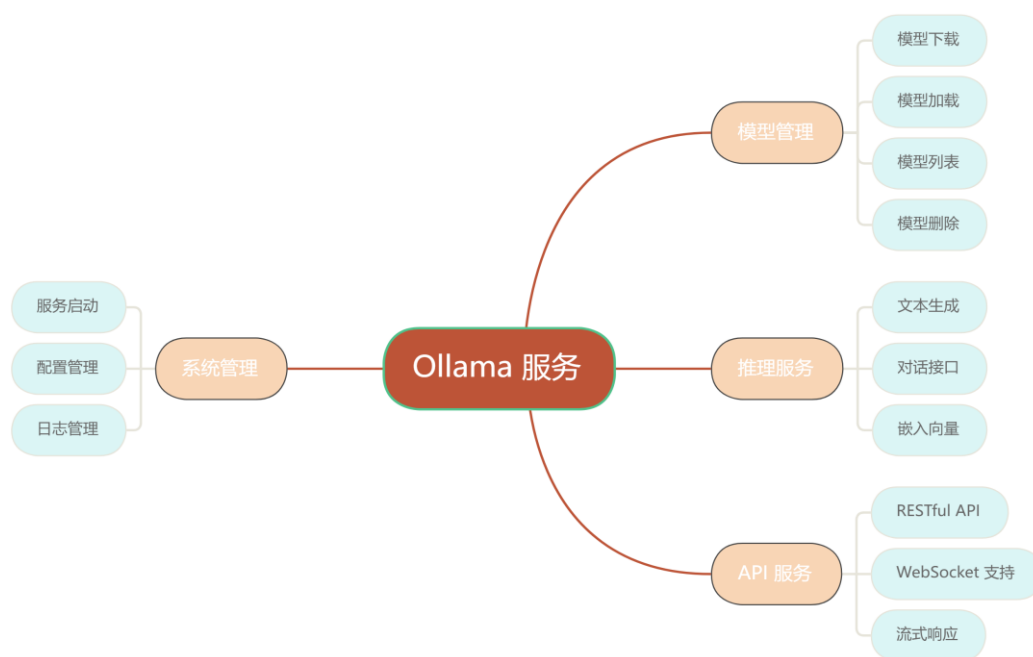


图 3-3 大模型引擎架构图

（四）数据流与安全性

在数据处理流与安全性层面，系统实施了全方位的安全加固。所有用户数据均在内存中处理，完成即释放，杜绝持久化存储带来的泄露风险；文件解析后立即释放资源，避免资源驻留。在传输层面，系统支持 HTTPS

加密与 JWT 令牌认证，确保通信安全。同时，提供可选的审计日志功能，记录关键操作元数据，满足合规性审计要求。

四、功能使用描述

“净语”系统的功能设计以简洁美观的用户交互与高效便捷的后台管理为核心，其完整的使用功能可划分为数据输入、策略执行、结果生成与系统维护四个方面。

（一）数据输入

系统提供了多元化的接入方式。用户可通过直观的 Web 界面直接输入待检文本，或上传 TXT、PDF、DOCX 格式的文档；对于图片形式的文本内容，系统集成 OCR 模块实现自动化的文字提取与识别，实现了对主流信息载体的全面覆盖，具体检测流程如图 4-1 所示。

除了访问 Web 界面，用户还可以通过基于 HTTP/HTTPS 协议的 API 接口直接调用文本检测(/detect/text)、文档检测(/detect/document)、词库管理(/word-libraries)、健康检查(/health)等核心功能。系统提供了标准化接口，以实现与第三方平台的无缝衔接。

（二）数据处理

系统创新性地引入了“规则匹配+大模型”的双引擎检测机制和“高效+严格”双模式检测策略，以灵活适配不同的应用场景与性能要求。高效模式优先调用规则引擎进行毫秒级快速筛查，专为实时评论、即时通讯等高并发、低延迟场景设计。严格模式则在规则初筛基础上，强制触发大语言模型对全部内容进行深度语义分析与上下文校验，确保对涉及高敏感性文本的判定精度。这种分级的策略设计，使用户能够根据任务特性，动

态权衡处理速度与检测深度。

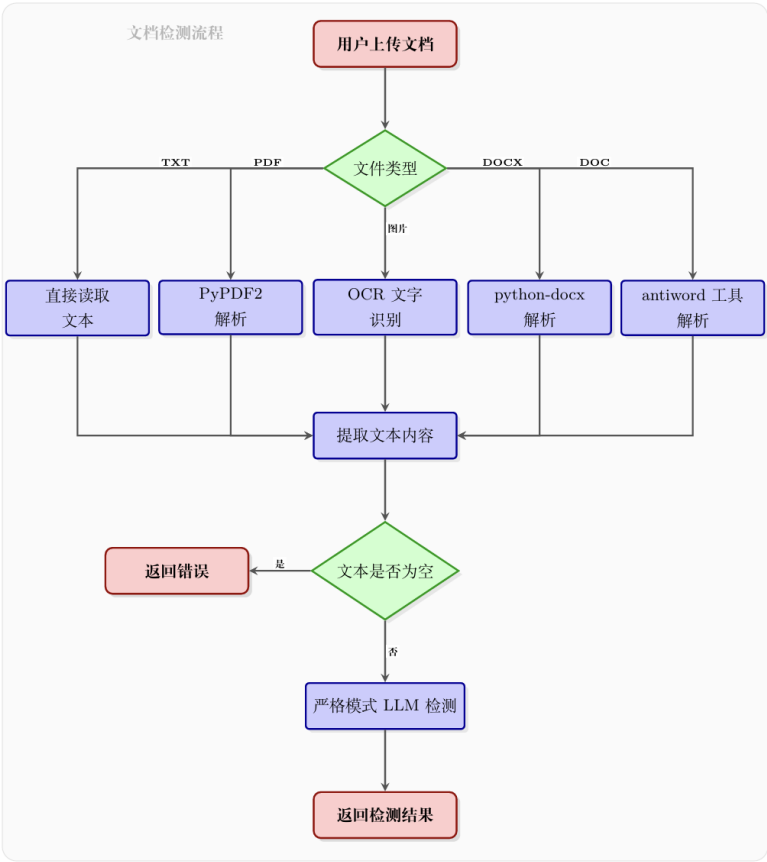


图 4-1 数据检测流程

（三）结果生成

经过对输入内容的检测，前端界面会简洁输出其触发的敏感词及最终判定结果，开发人员也可根据需要调取后台详细报告。同时系统后台预留了接口以支持敏感词定位、标记等其他功能拓展需求。

（四）系统维护

系统后台提供了健康检查机制，自动检测服务运行状态并返回状态码，以便于运维人员实时监控系统运行。系统前端为用户提供了便捷的敏感词库动态管理功能，管理员可对敏感词库执行创建、编辑、删除及导入/导出等操作，所有操作均可实时更新并生效于检测流程，确保系统能够迅捷响

应快速多变的网络用语与安全威胁，提升持续防护效力。

五、应用价值

“净语”系统的核心价值在于其突破了传统内容安全领域的“表层规则匹配”，实现了基于大语言模型的“深层语义理解”；同时创新解决了当前内容安全领域“效率与精度”难以兼顾的难题，提供了一个开源的、可拓展的、可快速部署的敏感词检测解决方案。

（一）从“表层规则匹配”到“深层语义理解”的突破

传统的敏感词检测工具往往采取规则匹配的方式，基于敏感词库对待审核内容进行字符层面的匹配。尽管可以通过扩充词库和更新算法实现对绝大多数敏感词及其变体的有效识别，但始终无法解决语义难题，例如“玩具枪”中的“枪”不应该被判定为敏感词。随着近几年自然语言处理领域的快速发展，我们借助大语言模型突破了敏感词检测的语义难题，能够结合上下文语义对敏感词进行更精准的识别。

（二）解决“效率与精度”难以兼顾的难题

基于规则匹配的检测方法效率较高，单个语句实时响应时间约为 5-10ms，但容易误报、漏报；基于大模型的检测方法可以有效解决误报、漏报问题，但效率较低，单个语句实时响应时间约为 300-500ms，连续相应时间约为 100-200ms。传统的基于规则匹配的方法和完全基于大模型的方法都存在“效率与精度”的矛盾，本系统通过建立规则引擎与 LLM 的协同判别的机制，在维持高实时性的同时，大幅提升了审核作业的综合效能。

（三）高度的灵活性与扩展性

“高效+严格”双模式策略使系统能够灵活适配不同场景，例如在实

时评论、即时通讯等高并发、低延迟场景，系统需要优先考虑敏感词检测的吞吐量和实时性；在政治性言论审查、重要舆情监控等高敏感场景中，系统需要优先考虑敏感词检测的准确率和召回率。本系统通过双模式策略能够在效率与精度之间做出动态调整，且模块化设计具有高拓展性，可以根据用户需要定制个性化的检测功能，以满足不同业务场景的差异化需求。

（四）快速部署能力与广泛应用前景

本项目采用 docker 容器化部署，解决了跨环境依赖问题，支持“一键部署”“开箱即用”，内置完整的敏感词检测功能、丰富的敏感词库（包含超过 7 万条敏感词）、完善的技术文档，支持用户快速部署和二次开发。

本项目可广泛应用于社交、政务、电商、教育、金融、直播、游戏等平台的内容合规性审核，针对文本、语音转文字、文案、聊天记录等各类内容实现高效自动检测，为各行业场景提供内容风险防控与安全保障。还可应用于网络舆情监控、政治性言论审查、涉密信息监测等高敏感场景，为相关部门提供智能化审查、监控服务。

综上所述，“净语”系统不仅通过技术创新解决了内容审核的关键难题，更从工程部署与实际应用出发，面向多应用场景展现了内容安全的核心竞争力。

六、创新点

（一）双引擎驱动，兼顾精准与高效

在实际场景中，大多数待审核内容属于正常内容，敏感内容占比极少。本系统采用“规则引擎+大语言模型”的协同判别架构，规则引擎基于 AC 自动机与 DFA 实现快速初筛，单个语句响应时间约为 5ms，确保快速过

滤大多数正常内容；大模型层则对初筛存疑内容进行深度语义解析，精准识别传统方法难以应对的谐音变体、语境伪装等复杂形态。二者协同工作，在保持高效率的同时，显著提升了系统的语义理解与深度检测能力。

（二）容器化部署，跨环境一键启动

本项目使用 `docker` 封装所有核心组件，编写了自动化部署脚本，可实现“克隆代码库、执行 `docker compose up`”的一键式启动，有效解决了跨环境依赖问题。此外，容器健康检查机制会实时监控服务状态，有效降低运维成本。这种设计既保证了开发、测试、生产环境的一致性，又为规模化部署和快速迭代提供了基础。

（三）模块化设计，支持功能拓展与灵活接入

系统采用模块化设计，将核心功能拆解为独立模块，分别为规则引擎模块、LLM 调用模块、文档解析模块、词库管理模块及前端交互模块，各模块通过标准化接口通信。该设计使系统具备较强的可扩展性，开发人员可根据需要快速修改或拓展功能模块；第三方系统可通过 API 快速接入检测能力，或通过挂载自定义词库目录实现个性化需求。模块间的低耦合特性也降低了维护成本，便于团队并行开发与迭代。

（四）双轨检测策略，灵活适配多场景

系统引入“高效模式”与“严格模式”双轨并行策略，使用户能够根据任务敏感性与实时性要求灵活调整检测强度。高效模式优先保障响应速度，适用于实时交互场景；严格模式则全面启用语义深度分析，确保高敏感内容识别的准确性。该设计兼顾了性能与精度之间的平衡，拓宽了系统的适用边界。

（五）多格式输入，一站式文档解析

系统集成多格式输入解析机制，支持实时文本、TXT/PDF/DOCX 文档以及 OCR 文本识别，实现了对主流数据格式的一站式无缝处理。有效解决了传统审核流程中因工具碎片化导致的操作复杂与效率低下问题，提升了系统在多媒体数据流场景下的整体兼容性与处理效率。

（六）敏感词库管理，实时动态更新

系统提供完善便捷的敏感词库管理功能，支持敏感词库导入、自定义词库、词条编辑等操作。词库变动可即时生效于检测流程，结合规则引擎的高效匹配能力，使系统具备快速响应新型敏感内容与动态威胁的能力，显著增强了系统的自适应性及持续防护效能。

七、核心团队人员介绍

指导老师：李永，武警工程大学北方云重点实验室副主任，职称教授，研究方向大模型识别，深度学习，武警信息化建设。

参赛队长：姜佳泽，武警工程大学研究生学员，研究方向：深度学习，行为识别。

参赛成员 1：王赛，武警河南总队在职干部、武警工程大学研究生学员，研究方向：作战数据保障。

参赛成员 2：程关潼，武警工程大学研究生学员，研究方向：作战信息通信保障。

参赛成员 3：胡玉强，武警工程大学研究生学员，研究方向：深度学习，行为跟踪。

参赛成员 4：张宇泽，武警工程大学本科学员，在读专业：应用心理

学。

八、核心成果展示

为评估“净语”系统的性能优势，我们设计了对比实验，将其与规则匹配模型在相同数据集上进行敏感词识别效果比对。其中，规则匹配模型由本研究所构建的“AC 自动机+DFA”模块独立运行，未引入大语言模型进行语义辅助判断。实验从直接敏感词、字符变体、谐音变体、上下文伪装、负面边界样本（容易误判的敏感内容）、正面边界样本（容易误判的正常内容）6个维度对识别准确率和召回率进行系统评估，具体对比结果如图 8-1 所示。

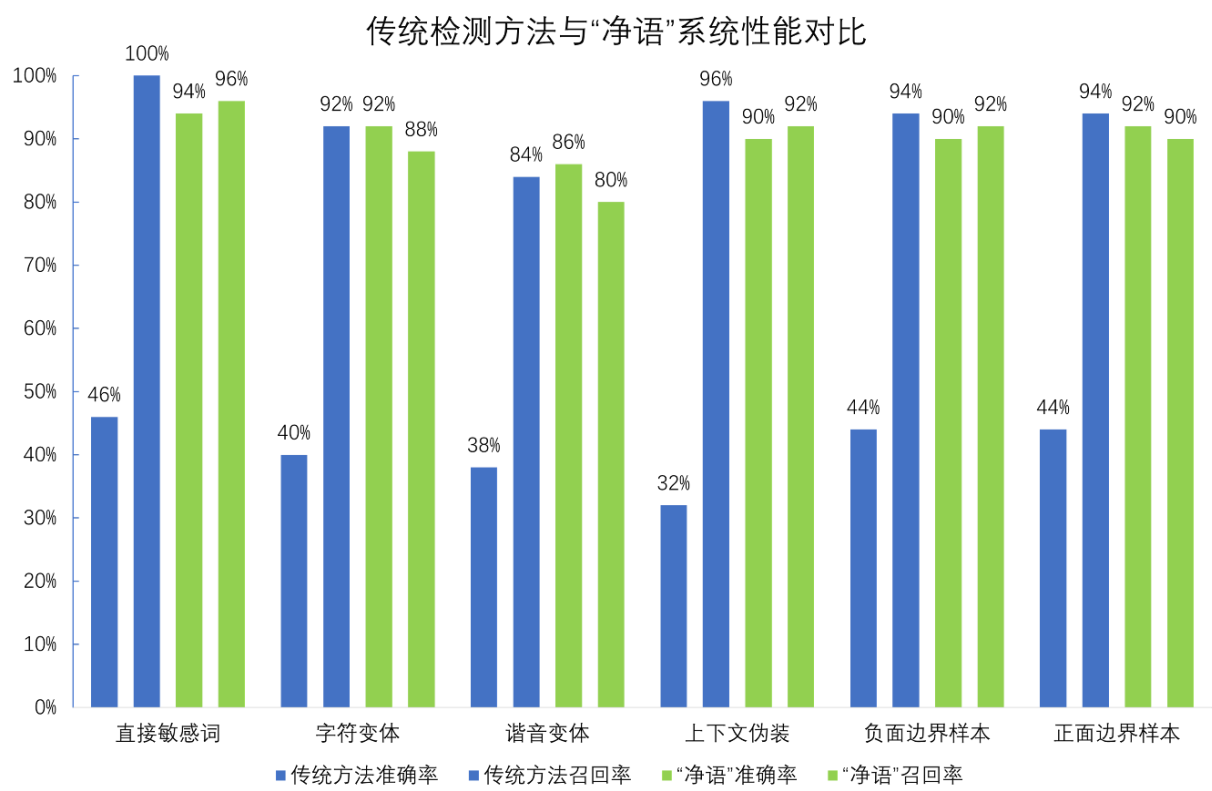


图 8-1 检测结果对比