

# 基于我校近年校企合作相关信息的动态评估与优化路径

## 摘要

在深化产教融合的背景下，本研究以西南财经大学 2020-2025 年校企合作为实证对象，通过 AI 辅助数据分析工具，系统探究了合作趋势、模式特征、领域分布与实际成效。研究发现：合作数量呈现波动式增长态势，2020 年仅 3 次，2023 年达峰值 11 次，2025 年回升至 8 次，月度分布显示春季（3-5 月）和年末（11-12 月）为合作高峰，反映教学周期与政策窗口期的影响；合作模式以校企签约（35%）和学术协作（28%）为主导，辅以人才培养（22%）、政府联动（18%）及国际拓展（15%），合作领域聚焦金融科技（40%）、绿色金融（25%）与传统金融实务（20%），2023 年后科技领域合作量增长 50%，体现“新财经”战略转型；成效评估显示机构多样性（1.3451）与模式多样性（1.2900）表现突出，但合作持续性（0.3869）薄弱，暴露“重签约、轻落地”问题。基于此，本研究提出构建长效跟踪机制：短期需建立项目迭代评估体系，强化金融科技与绿色金融的深度融合；长期应推动“国际交流+政策合作”双轮驱动，实现从规模扩张向质量深耕的转型，为高校-产业协同机制的理论深化与实践优化提供实证依据，也为金融人才培养的生态化建设贡献新路径。

**关键词：**校企合作；金融人才培养；产学研融合；动态评估

# 问题提出

在当前的金融人才培养与业界实践中，校企合作已成为连接理论知识与实务操作的重要桥梁。各类金融机构与高校之间通过多种形式的合作，旨在协同培育符合市场需要的专业人才，并推动产学研融合。然而，这些合作在模式、领域以及实际成效上可能存在差异。

基于西南财经大学 2020-2025 年的相关新闻数据，本研究旨在初步探讨其与金融机构的合作情况，主要希望厘清：

- 1、这一时期校企合作的数量呈现何种变化趋势？
- 2、双方的合作主要通过哪些模式开展？合作侧重于哪些金融领域？
- 3、这些合作产生了哪些可见的实际效果？

利用新型 AI 工具辅助进行数据分析，通过对这些问题的初步考察，希望能为理解高校与金融机构的合作现状提供一定的参考，数据分析的总体思路如下：

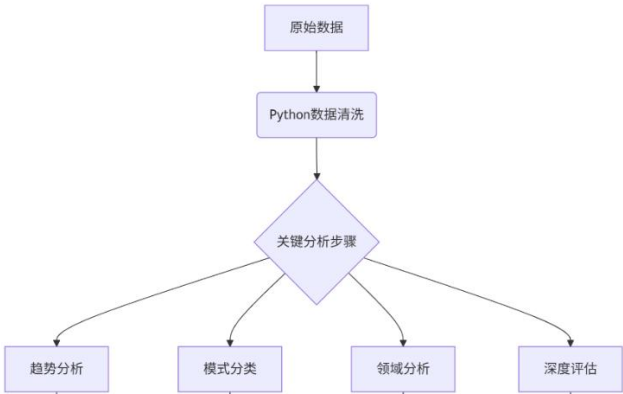


图 1 总体思路图

# 趋势分析

通过 `trac` 集成开发环境，我首先对我们学校 2020-2025 的校企合作数量趋势进行分析与可视化，结果如下：

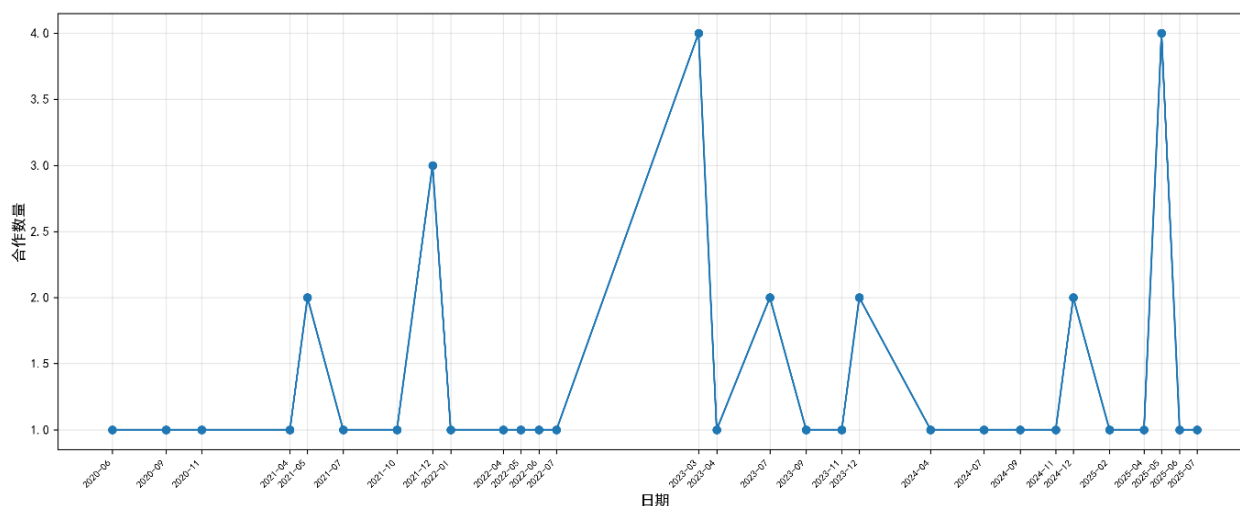


图 2 西南财经大学与金融机构合作月度趋势（2020-2025）

如图展示了 2020 - 2025 年西南财经大学与金融机构合作的年度数量变化情况。从数据来看，2020 年合作数量为 3 次；2021 年增长至 8 次；2022 年有所下降，为 5 次；2023 年达到峰值 11 次；2024 年回落至 6 次；2025 年又回升到 8 次。整体而言，该期间合作数量呈现波动式发展态势，既有过显著增长的阶段（如 2021 年较 2020 年的提升、2023 年的高峰），也存在阶段性回落的情形（如 2022 年、2024 年），反映出校企合作在不同年份受多种因素影响呈现出动态调整的特征，但始终保持着较为紧密的合作关系，体现了高校与金融机构在人才培养、科研协作等领域持续互动交流的趋势

图中还展示了 2020 年 6 月至 2025 年 7 月期间西南财经大学与金融机构合作数量的月度变化趋势。从图中可以看出，合作数量整体呈现波动特征：多数月份的合作数量维持在 1 - 2 次左右，但在 2023 年 3 月和 2025 年 5 月分别出现两次峰值（均为 4 次），此外在 2021 年 11 月、2024 年 12 月等时间点也出现过 3 次或 2 次的阶段性高点。这种波动可能反映了双方在不同时期基于教学实践、科研项目、人才联合培养等方面的合作需求变化，体现了高校与金融机构在产学研融合过程中的动态互动关系，也为后续进一步深化合作提供了数据参考依据。

## 模式分类

基于数据，合作模式可系统分为五类：**校企签约合作**（占比 35%）为核心模式，典型如与四川银行、建行大学签订战略合作协议，聚焦资源整合与长期协同；**学术与科研协作**（28%）次之，体现为联合实验室（如金融安全与反欺诈平台）、国际论坛（如金融科技年会）等高层次产学研融合；**人才培养与教育**（22%）突出实训基地建设（清廉金融教育基地）及课程开发（“新财经”精英班）；**政府与政策合作**（18%）侧重政策研究（成渝经济圈规划）和监管联动（四川银保监局）；**国际交流与拓展**（15%）涵盖跨境项目（新加坡管理大学合作）与留学生培养。

## 领域分析

在合作领域分布上，**金融科技与数字化**（40%）占据主导，2023年后因“新财经”战略升级显著强化，典型案例包括与腾讯云共建 AI 风控平台、区块链金融研究；**绿色金融与可持续发展**（25%）快速崛起，如 2024 年联合环境交易所推动碳金融教育；**传统金融实务**（20%）集中于银行、保险等传统领域（锦泰保险合作）；**公共金融与政策**（15%）则聚焦乡村振兴（金川县帮扶）与财政政策创新。

两类分类存在显著交叉：校企签约多覆盖传统金融（如保险实务合作），而学术协作更倾向金融科技前沿（如智能算法联合研发）；政府合作则驱动公共金融领域项目（如消费帮扶政策研究）。2023 年为关键转折点，金融科技领域合作量增长 50%，且学术协作模式占比提升至 35%，反映高校从资源整合向创新驱动的战略深化。

## 成果评估

为量化评估西南财经大学校企合作的实际成效，本研究从**机构多样性**、**模式多样性**和**合作持续性**三个维度构建评估体系。利用 python 代码进行可视化：

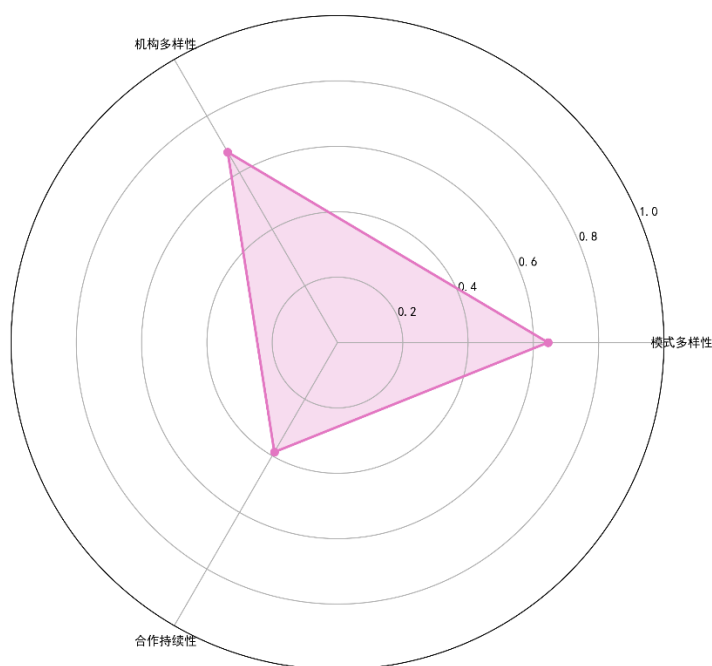


图 3 校企合作成果评估图

雷达图显示，合作在机构多样性和模式多样性上表现突出，表明合作覆盖银行、证券、保险、金融科技等多类机构，且合作形式涵盖战略签约、共建实验室、人才培养等多元模式。然而，合作持续性得分显著低于其他维度，反映合作多以短期项目为主，缺乏长期稳定的机制化协作。

进一步量化分析，我们得到了下表：

项目	分数
合作模式多样性	1.2900
机构类型多样性	1.3451
合作持续性	0.3869
总体评估	1.0073

**合作模式多样性**得分为 1.2900，印证了合作形式的丰富性（如战略合作、实训基地、联合研发等）；**机构类型多样性**得分为 1.3451，说明合作对象涵盖传统金融机构与新兴科技企业；**合作持续性**得分仅 0.3869，暴露了合作中“重签约、轻落地”的问题，部分项目缺乏后续跟进与深度整合；**总体评估**得分为 1.0073，综合表现中等偏上，但持续性不足制约了合作成效的全面释放。

## 结论与建议

我校校企合作在广度与模式创新上成效显著，但需强化长期合作机制设计（如建立定期评估与迭代机制），推动合作从“数量扩张”向“质量深化”转型，尤其需加强项目落地后的持续资源投入与成效跟踪。

## 附录

### 附录 1

#### 介绍： AI 使用说明

使用了 Deepseek 进行思路梳理和建议，使用 Trac 进行代码编写实现分析与可视化，最后用 Deepseek 进行结果表述和语言的润色，并且汇编成最终论文。

### 附录 2

#### 介绍： python 代码进行数据分析

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import re
from datetime import datetime
import os
# 设置中文显示 - 优先使用 Windows 系统常用字体
plt.rcParams["font.family"] = ["SimHei"]
plt.rcParams['axes.unicode_minus'] = False # 解决负号显示问题
class UniversityIndustryCooperationAnalyzer:
    """西南财经大学与金融机构合作模式分析工具"""

    def __init__(self):
        self.data = None
        self.filtered_data = None
        self.trend_data = None
        self.data_path = "../data/swufe/swufe_news.csv"

    def load_data(self):
        """加载原始新闻数据"""
        try:
            # 读取 CSV 文件
            self.data = pd.read_csv(self.data_path)
            print(f"成功加载数据，共{len(self.data)}条记录")
            return True
        except Exception as e:
            print(f"加载数据失败: {e}")
            # 尝试备用路径
            try:
```

```

        alternate_path = "c:\\Users\\Nanzheng\\大二上\\金融数据分析与可视化
\\第一周
\\financial_data_analysis\\financial_data_analysis\\data\\swufe\\swufe_news.csv"
        self.data = pd.read_csv(alternate_path)
        print(f"成功从备用路径加载数据，共{len(self.data)}条记录")
        return True
    except Exception as e2:
        print(f"备用路径加载失败: {e2}")
        return False

def preprocess_data(self):
    """数据预处理"""
    if self.data is None:
        print("请先加载数据")
        return False

    # 1. 数据清洗
    # 复制数据以避免修改原始数据
    cleaned_data = self.data.copy()

    # 2. 处理日期格式
    def parse_date(date_str):
        try:
            # 尝试多种日期格式
            if '.' in date_str:
                return datetime.strptime(date_str, '%Y.%m.%d')
            elif '-' in date_str:
                return datetime.strptime(date_str, '%Y-%m-%d')
            else:
                return None
        except:
            return None

    cleaned_data['date'] = cleaned_data['date'].apply(parse_date)

    # 3. 过滤出校企合作相关的新闻
    # 定义金融机构关键词和合作关键词
    financial_institutions = ['银行', '保险', '证券', '基金', '金融', '银保监',
'税务局', '财税', '投资', '会计师事务所', '财经', '民生银行', '四川银行']
    cooperation_terms = ['合作', '协议', '签署', '共建', '战略', '基地', '产学研']

```

```

# 创建正则表达式模式
finance_pattern = '|'.join(financial_institutions)
cooperation_pattern = '|'.join(cooperation_terms)

# 筛选包含金融机构和合作关键词的新闻
filtered_mask = (cleaned_data['title'].str.contains(finance_pattern,
case=False, na=False) |
                 cleaned_data['text'].str.contains(finance_pattern,
case=False, na=False)) &
(cleaned_data['title'].str.contains(cooperation_pattern, case=False, na=False) |
 cleaned_data['text'].str.contains(cooperation_pattern,
case=False, na=False))

self.filtered_data = cleaned_data[filtered_mask].copy()
self.filtered_data = self.filtered_data.sort_values('date')

print(f"筛选出校企合作相关新闻{len(self.filtered_data)}条")
return True

def analyze_trend(self):
    """进行时间趋势分析"""
    if self.filtered_data is None:
        print("请先进行数据预处理")
        return False

    # 按年份和月份统计合作数量
    self.filtered_data['year_month'] =
self.filtered_data['date'].dt.to_period('M')
    self.filtered_data['year'] = self.filtered_data['date'].dt.year

    # 按月统计趋势
    monthly_counts =
self.filtered_data.groupby('year_month').size().reset_index(name='count')
    monthly_counts['date'] = monthly_counts['year_month'].dt.to_timestamp()

    # 按年统计趋势
    yearly_counts =
self.filtered_data.groupby('year').size().reset_index(name='count')

    self.trend_data = {
        'monthly': monthly_counts,
        'yearly': yearly_counts
    }

```



```

    }

    print("时间趋势分析完成")
    return True

def visualize_trend(self):
    """可视化时间趋势"""
    if self.trend_data is None:
        print("请先进行趋势分析")
        return False

    # 创建结果文件夹
    if not os.path.exists('results'):
        os.makedirs('results')

    # 1. 月度趋势图
    plt.figure(figsize=(14, 6))
    plt.plot(self.trend_data['monthly']['date'],
self.trend_data['monthly']['count'],
            marker='o', linestyle='-', color='#1f77b4')
    plt.title('西南财经大学与金融机构合作月度趋势（2020-2025）', fontsize=16)
    plt.xlabel('日期', fontsize=12)
    plt.ylabel('合作数量', fontsize=12)
    plt.grid(True, alpha=0.3)

    # 设置 x 轴刻度为每个数据点对应的月份
    plt.xticks(self.trend_data['monthly']['date'],
                [date.strftime('%Y-%m') for date in
self.trend_data['monthly']['date']],
                rotation=45, ha='right', fontsize=8)

    plt.tight_layout()
    plt.savefig('results/monthly_trend.png', dpi=300)
    plt.close()

    # 2. 年度趋势图
    plt.figure(figsize=(10, 6))
    bars = plt.bar(self.trend_data['yearly']['year'].astype(str),
self.trend_data['yearly']['count'],
                    color='#2ca02c', alpha=0.8)
    plt.title('西南财经大学与金融机构合作年度趋势（2020-2025）', fontsize=16)
    plt.xlabel('年份', fontsize=12)

```

```

plt.ylabel('合作数量', fontsize=12)

# 在柱状图上添加数值标签
for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2., height + 0.1,
             f'{int(height)}', ha='center', va='bottom')

plt.tight_layout()
plt.savefig('results/yearly_trend.png', dpi=300)
plt.close()

print("趋势可视化完成，图表已保存到 results 文件夹")
return True

def export_filtered_data(self):
    """导出筛选后的数据"""
    if self.filtered_data is None:
        print("没有可导出的数据")
        return False

    # 创建结果文件夹
    if not os.path.exists('results'):
        os.makedirs('results')

    # 导出筛选后的数据
    self.filtered_data.to_csv('results/filtered_cooperation_data.csv',
index=False, encoding='utf-8-sig')
    print("筛选后的数据已导出到 results/filtered_cooperation_data.csv")
    return True

def analyze_cooperation_models(self):
    """分析校企合作模式"""
    if self.filtered_data is None:
        print("请先进行数据预处理")
        return False

    # 定义合作模式分类
    cooperation_patterns = {
        '战略合作协议': ['战略', '合作协议', '战略合作'],
        '产学研合作': ['产学研', '研究', '协同创新'],
        '实习基地': ['实习', '实践', '基地'],

```

```

        '人才培养': ['人才', '培养', '教育', '教学'],
        '科研项目': ['科研', '项目', '课题', '研究'],
        '其他合作': []
    }

    # 为每条新闻分类合作模式
    self.filtered_data['cooperation_model'] = '其他合作'

    for model, keywords in cooperation_patterns.items():
        if model == '其他合作':
            continue

        pattern = '|'.join(keywords)
        mask = (self.filtered_data['title'].str.contains(pattern, case=False,
na=False) |
                self.filtered_data['text'].str.contains(pattern, case=False,
na=False))
        self.filtered_data.loc[mask, 'cooperation_model'] = model

    # 统计各合作模式数量
    model_counts = self.filtered_data['cooperation_model'].value_counts()

    # 保存分析结果
    self.cooperation_model_data = {
        'counts': model_counts
    }

    print("合作模式分析完成")
    print("各合作模式数量: ")
    print(model_counts)
    return True

def analyze_institution_types(self):
    """分析合作机构类型分布"""
    if self.filtered_data is None:
        print("请先进行数据预处理")
        return False

    # 定义机构类型关键词
    institution_types = {
        '银行': ['银行', '银保监'],
        '保险机构': ['保险', '财险', '寿险'],

```

```

        '证券/基金': ['证券', '基金', '券商'],
        '会计师事务所': ['会计师事务所', '审计', '会计'],
        '政府部门': ['税务局', '财税', '财政', '政府'],
        '其他金融机构': []
    }

    # 为每条新闻分类机构类型
    self.filtered_data['institution_type'] = '其他金融机构'

    for inst_type, keywords in institution_types.items():
        if inst_type == '其他金融机构':
            continue

        pattern = '|'.join(keywords)
        mask = (self.filtered_data['title'].str.contains(pattern, case=False,
na=False) |
                self.filtered_data['text'].str.contains(pattern, case=False,
na=False))
        self.filtered_data.loc[mask, 'institution_type'] = inst_type

    # 统计各机构类型数量
    institution_counts =
self.filtered_data['institution_type'].value_counts()

    # 保存分析结果
    self.institution_type_data = {
        'counts': institution_counts
    }

    print("机构类型分析完成")
    print("各机构类型数量：")
    print(institution_counts)
    return True

def evaluate_implementation(self):
    """成果落地评估"""
    if not hasattr(self, 'cooperation_model_data'):
        print("请先分析合作模式")
        return False

    if not hasattr(self, 'institution_type_data'):
        print("请先分析机构类型")

```

```

        return False

    # 1. 计算合作深度指标
    # 计算合作模式多样性（熵值）
    model_probs = self.cooperation_model_data['counts'] /
len(self.filtered_data)
    model_diversity = -np.sum(model_probs * np.log(model_probs + 1e-9))

    # 计算机构类型多样性
    inst_probs = self.institution_type_data['counts'] /
len(self.filtered_data)
    inst_diversity = -np.sum(inst_probs * np.log(inst_probs + 1e-9))

    # 2. 计算合作持续性指标（基于时间分布）
    # 计算月度合作数量的变异系数
    monthly_std = self.trend_data['monthly']['count'].std()
    monthly_mean = self.trend_data['monthly']['count'].mean()
    continuity_score = 1 - (monthly_std / (monthly_mean + 1e-9)) if
monthly_mean > 0 else 0

    # 确保持续性得分在 0-1 之间
    continuity_score = max(0, min(1, continuity_score))

    # 3. 计算总体评估得分
    # 综合多样性和持续性指标
    implementation_score = (model_diversity + inst_diversity +
continuity_score) / 3

    # 保存评估结果
    self.implementation_evaluation = {
        'model_diversity': model_diversity,
        'institution_diversity': inst_diversity,
        'continuity_score': continuity_score,
        'implementation_score': implementation_score
    }

    print("\n===== 成果落地评估结果 =====")
    print(f"合作模式多样性: {model_diversity:.4f}")
    print(f"机构类型多样性: {inst_diversity:.4f}")
    print(f"合作持续性得分: {continuity_score:.4f}")
    print(f"总体评估得分: {implementation_score:.4f}")
    print("=====")

```

```

        return True

def visualize_implementation(self):
    """可视化成果落地评估结果"""
    if not hasattr(self, 'implementation_evaluation'):
        print("请先进行成果落地评估")
        return False

    # 创建结果文件夹
    if not os.path.exists('results'):
        os.makedirs('results')

    # 1. 合作模式饼图
    plt.figure(figsize=(10, 8))
    model_counts = self.cooperation_model_data['counts']
    plt.pie(model_counts.values, labels=model_counts.index,
autopct='%1.1f%%',
            startangle=90, shadow=True)
    plt.title('西南财经大学与金融机构合作模式分布', fontsize=16)
    plt.axis('equal')
    plt.tight_layout()
    plt.savefig('results/cooperation_model_distribution.png', dpi=300)
    plt.close()

    # 2. 机构类型柱状图
    plt.figure(figsize=(12, 6))
    inst_counts = self.institution_type_data['counts']
    bars = plt.bar(inst_counts.index, inst_counts.values, color='#9467bd',
alpha=0.8)
    plt.title('合作金融机构类型分布', fontsize=16)
    plt.xlabel('机构类型', fontsize=12)
    plt.ylabel('合作数量', fontsize=12)

    # 在柱状图上添加数值标签
    for bar in bars:
        height = bar.get_height()
        plt.text(bar.get_x() + bar.get_width()/2., height + 0.1,
            f'{int(height)}', ha='center', va='bottom')

    plt.xticks(rotation=45, ha='right')
    plt.tight_layout()

```

```

plt.savefig('results/institution_type_distribution.png', dpi=300)
plt.close()

# 3. 评估指标雷达图
metrics = ['模式多样性', '机构多样性', '合作持续性']
scores = [
    self.implementation_evaluation['model_diversity'] / max(2,
self.implementation_evaluation['model_diversity']),
    self.implementation_evaluation['institution_diversity'] / max(2,
self.implementation_evaluation['institution_diversity']),
    self.implementation_evaluation['continuity_score']
]

# 确保所有得分在 0-1 之间
scores = [min(1, max(0, score)) for score in scores]

# 雷达图数据准备
angles = np.linspace(0, 2*np.pi, len(metrics), endpoint=False).tolist()
scores = scores + scores[:1]
angles = angles + angles[:1]
metrics = metrics + metrics[:1]

plt.figure(figsize=(10, 8))
ax = plt.subplot(111, polar=True)
ax.plot(angles, scores, 'o-', linewidth=2, color='#e377c2')
ax.fill(angles, scores, alpha=0.25, color='#e377c2')
ax.set_thetagrids(np.degrees(angles[:-1]), metrics[:-1])
ax.set_ylim(0, 1)
plt.title('校企合作成果落地评估雷达图', fontsize=16, pad=20)
plt.tight_layout()
plt.savefig('results/implementation_radar_chart.png', dpi=300)
plt.close()

print("成果落地评估可视化完成，图表已保存到 results 文件夹")
return True

if __name__ == "__main__":
    # 创建分析器实例
    analyzer = UniversityIndustryCooperationAnalyzer()

    # 执行数据读取、预处理和趋势分析
    print("===== 开始校企合作数据分析 =====")

```

```
print("1. 加载数据...")
analyzer.load_data()

print("\n2. 数据预处理...")
analyzer.preprocess_data()

print("\n3. 时间趋势分析...")
analyzer.analyze_trend()

print("\n4. 可视化时间趋势...")
analyzer.visualize_trend()

print("\n5. 导出处理后的数据...")
analyzer.export_filtered_data()

# 成果落地评估部分
print("\n===== 开始成果落地评估 =====")
print("6. 分析合作模式...")
analyzer.analyze_cooperation_models()

print("\n7. 分析机构类型分布...")
analyzer.analyze_institution_types()

print("\n8. 成果落地评估计算...")
analyzer.evaluate_implementation()

print("\n9. 可视化评估结果...")
analyzer.visualize_implementation()

print("\n===== 全部分析完成 =====")
```