# IT创业实践-“科政明”政策推送平台项目规划书

# 第一章 需求分析

政策是传递政府决策、调整领域发展、促进社会进步的重要途经。而在科技蓬勃发展的现在，科技政策无疑是社会发展中重要的一环，习近平总书记深刻指出，那些抓住科技革命机遇走向现代化的国家，都是科学基础雄厚的国家。我们要不断夯实科技基础，筑牢科技自立自强的根基。科技基础既包括各类科技创新组织、科研设施平台、科学数据和文献期刊等“硬条件”，也包括科技政策与制度法规等“软环境”。因此，全国科技政策数据的研究具有重要意义。

如今信息爆炸式增长在使人们获取信息更加便利的同时，也带来了信息利用效率降低、阅读成本过高、查询难度增大等问题。其一，在面临新机遇、新挑战的当下，政策的颁布更加频繁，政策内容也更精细化，全部解读的阅读成本升高。其二，传统的政策信息推送方式多采用邮件、网站发布等方式，信息源分散，推送效果有限，需要个人主动寻找，耗费大量的时间和人力。

随着大数据和人工智能技术的迅猛发展， 传统自动文摘研究正朝着从抽取式摘要到生成式摘要的方向演化，从而达到生成更高质量的自然流畅的文摘的目的。近年来，深度学习技术逐渐被应用于生成式摘要研究中。生成式摘要技术提供了一种从海量文本中快速获取重要知识的手段，让用户快速了解该项政策的相关领域、目标人群，大大降低了阅读成本，从而具有广阔的应用前景。同时，基于用户偏好的精准推送可以将用户关心领域的内容精准展示给用户，减少用户搜索的时间，提高效率。

自动摘要生成及精准推送技术通过对信息的压缩和精炼，为提高知识获取效率提供了辅助手段。该技术的产生和发展使解决上述问题成为可能。

目前市场上已存在的政策推送类产品中，但大多存在一定缺陷：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 本产品 | 政策快报 | 政策通 | 掌上政策 |
| 推送功能 | 基于NLP政策实现的个性化推送 | 无推送，仅有最新政策显示 | 无推送 | 随机推送，未针对用户偏向优化 |
| 内容摘要 | 使用BERT模型自动生成政策摘要 | 无摘要 | 无摘要 | 无摘要 |
| 广告 | 无广告 | 首页存在广告 | 无广告 | 用户页存在广告 |
| 收费情况 | 完全免费 | 部分功能收费解锁 | 完全免费 | 部分功能收费解锁 |
| 政策更新 | 即时更新 | 即时更新 | 多个省市一年以上未更新 | 即时更新 |

综上，本团队结合科技政策浏览趋势的变化，收集了全国31个省级行政区域及部分重要城市科技部所发布的政策，共34000余条政策（政策通知21000余篇，政策解读9000余篇，最新资讯4000余篇）。对于获取到的数据使用NLP相关技术实现关键词提取，结合app使用过程中用户点击行为中生成的用户偏好，以此实现对于用户的个性化推送，及时发现和推送政策信息，保证政策信息的准确性和及时性，提高推送效果和用户满意度；使用BERT模型实现政策的自动化摘要生成，实现对政策文本的深度分析和挖掘，实现对政策信息的自动化处理和分析，帮助用户发现政策背后的规律和趋势，帮助用户了解政策为决策提供参考依据。本产品旨在利用深度学习技术实现科技政策的自动化摘要生成和精准推送，开发科技政策数据移动应用平台。协助科技政策传播，帮助用户解读政策，从而提高相关政策的执行效率，为政策的推广与执行提供帮助。

# 第二章 概要设计

## 2.1 功能模块

小程序功能模块设计具体结构如图1所示，主要包含以下：

（1）科技政策个性化精准推送模块根据用户选择的地区、政策发布时间区间，结合用户阅读记录进行精准推送。

（2）科技政策摘要生成模块在用户点击政策详情界面的“查看摘要”后，系统将根据相应算法生成摘要供用户快速了解政策。

（3）科技政策数据库搜索与查询模块在用户于“知识库”页面进入分类或直接搜索后，系统根据用户检索需求返回符合的科技政策信息。

（4）用户个人信息界面支持用户查询个人浏览历史、个人收藏信息以及获取最新系统通知。

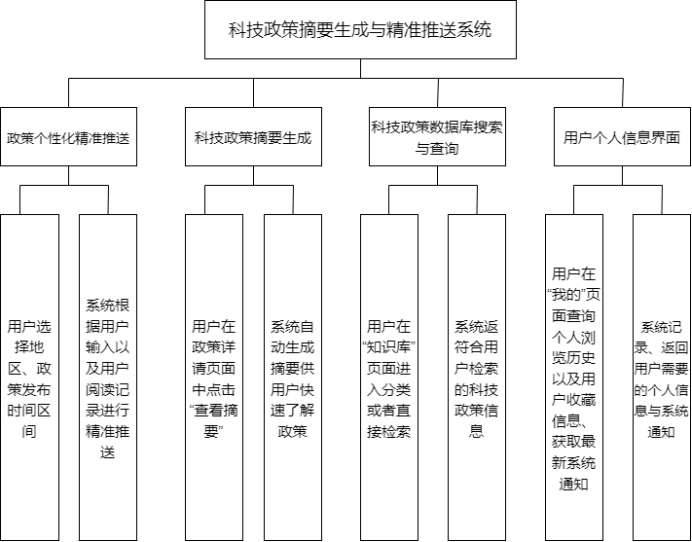


图1 功能模块

## 2.2 功能设计流程

小程序的架构主要分为三大部分，分别是用于用户操作的图形操作界面（小程序）、用于数据交互的Web服务器（Flask）和用于与Web服务器进行数据传输的模型及数据库。各项功能的实现主要基于上述三大部分。借助微信开发者工具平台，使用WXML（微信标记语言）、WXSS（微信样式表）以及JavaScript语言进行图形交互界面的设计与制作并用Python语言构建Web应用框架（Flask），用于实现小程序、模型及数据库之间的数据传输。

小程序的功能设计流程图如图所示。关键介绍多功能语义搜索以及个性化精准推送的实现设计。多功能语义搜索功能基于Flask所搭建的Web服务器，通过小程序—模型—数据库的数据交互实现。在小程序中通过点击搜索按钮进行问题搜索，发送request请求至服务器端。服务器端将问题语句传输给模型用于提取实体与关系。模型返回一对实体与关系名称至服务器，由服务器传递给数据库，进行查询操作后返回命中数据至小程序，最后进行结果展示。其中，Web服务器主要用于数据传输，模型主要用于提取问题语句中的实体与关系，数据库完成查询，小程序最终将结果展现反馈给用户。

小程序的功能设计流程图如图2所示，关键介绍个性化精准推送以及摘要生成功能的实现设计。该功能基于Flask所搭建的Web服务器，通过小程序—算法—数据库的数据交互实现。在小程序中点进推送界面，发送request请求至服务器端。服务器端将用户uid传递给数据库，进行查询操作后返回用户偏好信息。根据算法计算给出内容与用户偏好拟合度大于阈值的政策pid作为基于内容的推荐，并通过查询数据库获取最近浏览量最高的政策pid作为基于热点的推荐。将政策pid传递给数据库，获取政策原文返回至小程序，推送给用户。在知识库查询模块，基于生成式摘要模型返回政策全文及摘要信息。

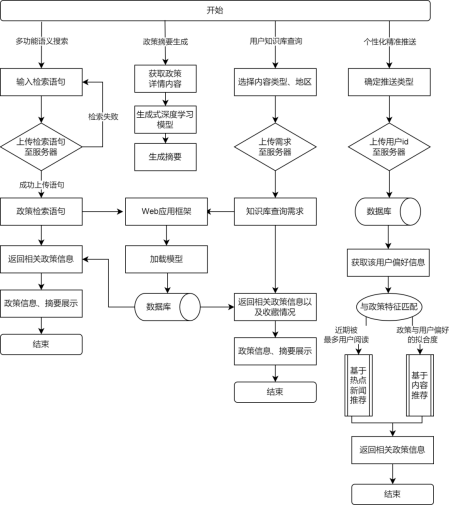


图2 功能设计流程图

# 第三章 详细设计

## 3.1 界面设计

本节将简单介绍小程序主要界面以及项目主要使用流程。

**1、科技政策知识库搜索与查询界面**

扫描二维码进入小程序主界面，具体布局如图 3 所示。颜色基调为蓝色（#2e50ca）、

米色（#faf5f5）、白色（#ffffff）三色，主体风格简约美观，冗余设计少，实用性强。

|  |  |
| --- | --- |
|  |  |

图3 主界面

界面上包括主搜索框和地区选择、政策分类按钮以及热门政策推荐栏，分别对应满足了用户对政策地区的限制需求，用户对特定政策类型的检索需求以及用户快速掌握当前政策热点的需求。

在查看分类功能中，用户选择自己感兴趣的内容分类后，可进入相应内容的界面，并在该界面进行进一步的细分（政策通知、政策解读、最新资讯），或者进行进一步的搜索框检索。该分类界面如图4所示。

|  |  |
| --- | --- |
|  |  |

图4 分类界面

在搜索功能中，用户在主搜索框输入检索语句后，点击软键盘上的搜索键或主界面上的搜索按钮，即可进行多功能语义搜索。若成功，则跳转到搜索结果页面；若输入空值或输入不合理的检索语句，则返回失败。 用户输入想要搜索的食品安全事件，在 Web 服务器的运行下，由小程序—模型—数据库进行数据交互并将最终结果呈现给用户。搜索结果页面如图5所示。

|  |  |
| --- | --- |
|  |  |

图5 检索界面

在任一界面点击某一政策后，便能进入当前政策详情界面。点击右下角的“查看摘要”按钮，则可进一步查看当前政策的摘要内容。界面如图6所示。

|  |  |
| --- | --- |
|  |  |

图6 政策详情界面

**2、科技政策个性化精准推送界面**

用户点击页面栏的推送按钮，便可进入到推送页面。在点击推送页面按钮的那一刻，系统便会根据用户的浏览记录与偏好信息生成推送，并在推送页面中按照分类显示（默认页面为"政策通知"页面）。用户可在页面中浏览自己想要的其他分类（"政策解读"或"最新资讯")，该界面如图7所示。

|  |  |
| --- | --- |
|  |  |

图7 推送界面

**3、用户个人信息界面**

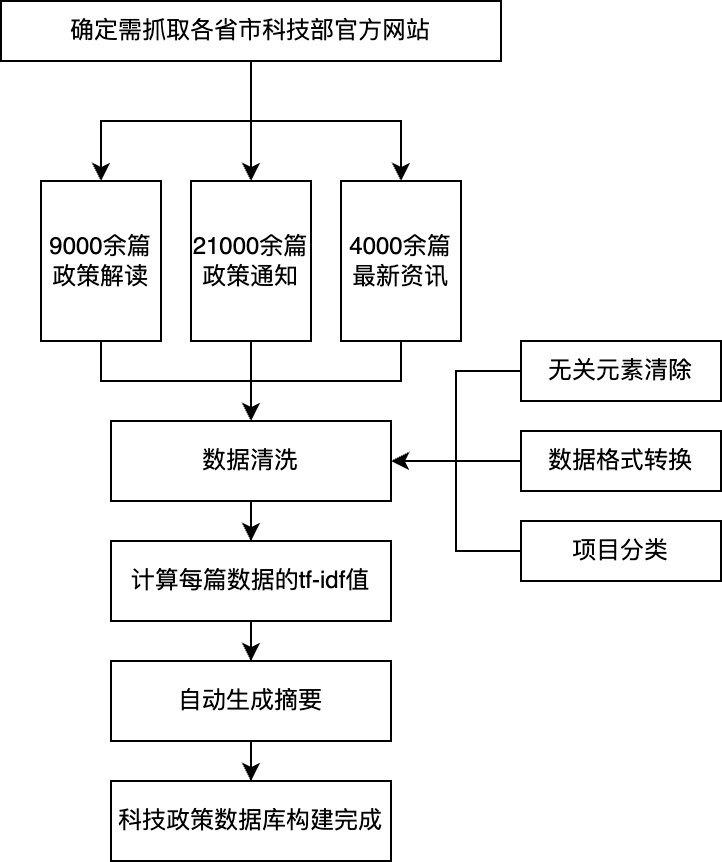
用户点击页面栏中的“我的”按钮，可以来到用户个人信息界面（如图8左所示），在该界面中，用户可以通过点击按钮，查看自己的浏览记录、收藏记录或程序推送的最新通知。（例如，用户收藏界面如图8右所示）。

|  |  |
| --- | --- |
|  |  |

图8 用户个人信息界面

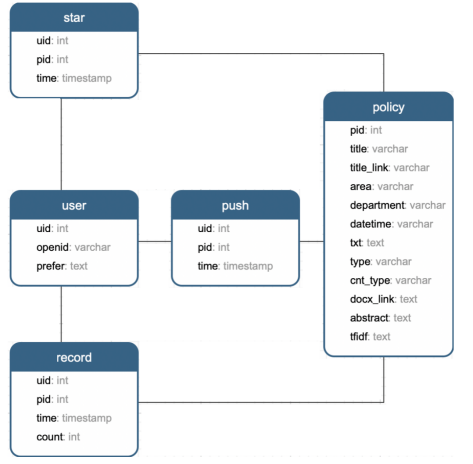
## 3.2数据库设计

本项目的数据处理流程如图，经过前期调研确定需抓取各省市科技部官方网站（见附件），爬取2022年1月至今共计xx篇科技政策及xx篇政策解读，经过数据清洗，最终共获取数据xx篇，在此基础上人工对语料进行部分属性标注，通过python计算每篇数据的tf-idf值，获取其关键词，并通过BERT模型自动生成摘要，最终形成了本项目所使用的科技政策数据库，基于此进行算法设计。



图x 数据处理流程

数据存储主要使用Python语言引入SQL对象关系映射器，实现Python程序与关系数据库（MySQL）的通讯，主要存储实体关系信息。主要实体及其属性如下图所示。

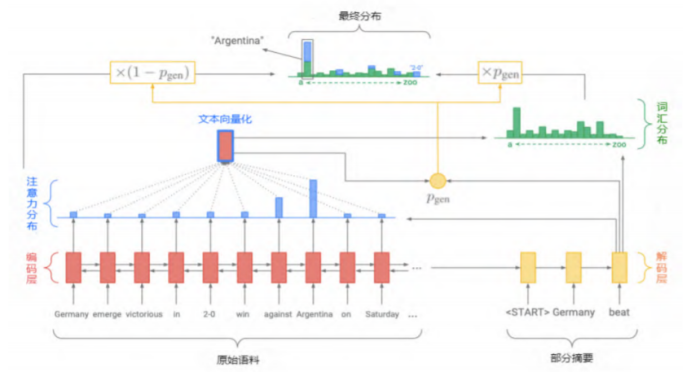


## 3.3关键技术

### 1.摘要生成算法

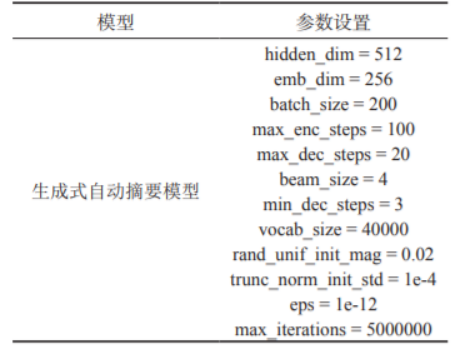
目前，自动摘要方法按生成方式主要分为抽取式自动摘要和生成式自动摘要，抽取式自动摘要主要应用关键词句排序的思想，而生成式自动摘要更多是基于深度学习模型来完成，本项目使用的生成式自动摘要模型，参考了基于指针生成网络构建的面向中文的Text-SummarizerPytorh-Chinese模型及其思路。

指针生成网络（pointer-generator network）的自动摘要任务原理如下图所示。该模型能够通过自注意力机制集中于文本中的重要词汇，并由此生成新词汇。它不同于抽取式自动摘要按词频和簇确定关键词，再通过关键词对所在句打分，分数排序确定最终生成摘要的句子来生成摘要的方式，而是权衡词表中词汇的概率、词汇分布以及注意力分布来确定候选词的权重并获得最终分布情况。



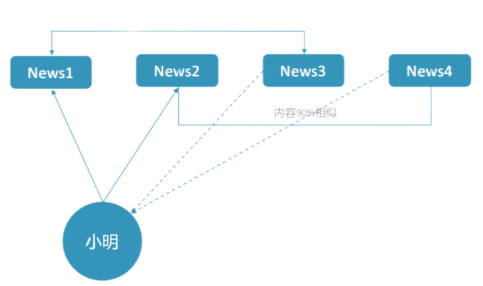
生成式自动摘要模型构建过程主要包括以下步骤：语料获取；待摘要文本预处理；预训练模型构建；生成式自动摘要模型训练；根据训练过程及结果调整参数并进行迭代训练；根据最终模型生成摘要。

本项目模型参数设置如下：



### 2.基于内容的推送算法

为了让用户更有效的获取自己想要了解的政策，推荐算法是最重要的一环，本项目使用的基于内容的推送算法的核心思想是通过用户已浏览内容的特征，推荐与其相似的其他内容。



算法的实现主要分为以下几步：

（1）获取文本的关键词

文本类的数据，本身可以从文本特征几个方面去提取它的特征信息，进而将不同的文本间的特征信息进行比较，本项目选择以关键词作为判断政策相似程度的依据。

本项目使用TF-IDF评估一字词对于一个文件集或一个语料库中的其中一份文件的重要程度。该算法下字词的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降。

词频(TF) = 某个词在文章中出现的次数/文章的总词数 ，

逆文档频率(IDF) = log ( 语料库的文档总数/(包含该词的文档数+1))

TF-IDF=词频(TF)\*逆文档频率(IDF)

（2）构建用户偏好

从用户历史浏览记录里挖掘用户的喜好，在数据库中为每个用户维持一个关键词列表，当用户浏览过某个政策后，利用TF-IDF算法提取出政策中的K个关键词及对应的TF-IDF值，并将它们存入用户相应的关键词列表中。

（3）用户偏好实时更新

当用户的关键词列表达到规定的长度后，用户继续浏览政策时会对其偏好列表进行更新，将列表中存在的TF-IDF值进行叠加，并将已有关键词与政策关键词共同进行排序，取最前列的n个词作为新的用户偏好表。

为防止用户因前期大量浏览某类政策后使其个别偏好关键词数值过高难以更替，需要定期对偏好列表进行衰减，本项目在每天24:00时对用户的偏好关键词对应tf-idf乘以系数0.1，经过测试该系数可以实现较好的效果。

（4）新闻内容与用户喜好拟合度计算

有了用户的喜好关键词列表{keyword1:value1,keyword2:value2……}，以及某条新闻的关键词列表{nkeyword1:nvalue1,nkeyword2:nvalue2……}，将两个表的键匹配与值的运算，若有相同的键，则值相乘，多个相同键的值乘积累加；若无相同的键，值记为0，对于所有政策该拟合度，将拟合度最高的20条推送给用户。

# 第四章 测试报告

本项目在试运行阶段对小程序展开了功能性测试和其他测试，功能性测试以系统响应时

间为指标，主要对小程序的各部分检索功能的运行速度进行了功能性测试，测试效果良好；其他测试主要包括系统可用性、安全性、方便性以及扩展性四个部分。

**1. 功能性测试**

表

|  |  |  |
| --- | --- | --- |
| 序号 | 语义检索 | 精准推送 |
| 0 | 0.323 | 0.104 |
| 1 | 0.213 | 0.21 |
| 2 | 0.204 | 0.156 |
| 3 | 0.209 | 0.145 |
| 4 | 0.288 | 0.137 |
| 5 | 0.222 | 0.178 |
| 6 | 0.221 | 0.155 |
| 7 | 0.27 | 0.144 |
| 8 | 0.248 | 0.145 |
| 9 | 0.299 | 0.189 |
| 平均 | 0.250 | 0.156 |

**2. 其他测试**

在可用性上，目前系统所有功能均可顺畅使用，检索返回结果基本准确。

在安全性上，测试用户均没有发现个人隐私信息被盗取、被利用、钱财损失等问题。

在方便性上，基于用户规模庞大的微信客户端，在小程序正式上线后，用户通过扫描小程序二维码即可进入使用，界面设计美观，实用性强，测试用户反馈良好。

在扩展性上，本系统在手机移动端扩展性好，基于目前数据库，小程序仍可继续增加或完善功能，如扩大政策数据量，扩大政策类型等。

# 第五章 安装及使用

本项目的小程序已布置到阿里云服务器上，但由于仍处于试运行阶段，故未上线发行，目前只有项目成员和体验成员才能使用体验版小程序。

**第六章 项目总结**

本项目以提高企业科技政策了解效率为出发点，针对企业了解政策操作不够便捷、消息不够准确等问题，通过设计优化NLP模型，设计科技政策自动化摘要生成与精准推送小程序，同时结合到爬取的文献数据、政策数据，进行清洗加工，将加工好的信息分类展示到小程序中，完善了小程序的功能，1.将零散杂乱无章的信息序化、系统化、一站式地展现给公众。本项目对科技政策自动化便利化精准推送具有较强的应用价值，对企业快速了解产品所需的相关政策有积极意义，同时也能为用户解读科技政策提供决策支持和技术支撑。在项目的实施过程中，我们首先查阅了相关资料，了解到我国的公共政策执行产生了很多的新问题，我们认识到为确保公共政策执行效率与成效的重要性，实现公共政策目标，应必须重视政策执行状况不佳的原因与完善路径问题，于是我们定题为科技政策摘要生成与精准推送。接着小组三人分工协作，负责小程序架构。一人负责前端设计，借助于微信开发者工具平台，使用 WXML（微信标记语言）、WXSS（微信样式表）以及 JavaScript 语言进行图形交互界面的设计；一人负责搭建数据库，构建模型并对其优化；另外一人负责爬取数据并连接前后端，使用Python 语言构建 Web 应用框架（Flask），用于实现小程序、模型及数据库之间的数据传输。在整个过程中，小组成员均参与数据清洗工作，参与小程序的整体功能设计。在项目实施过程中，我们遇到了不少问题：

1. 科技政策语料较为复杂，数据清洗过程比较困难为实现科技政策摘要生成的准确性，科技政策语料数据量庞大，因此数据清洗阶段任务量较大。
2. 小程序页面响应较慢将小程序数据库挂载到服务器上后，加载速度较慢，导致推送速度较慢。通过小组成员的共同商讨后，决定将数据库移至本地。

为了能够进一步提高该科技政策摘要生成与推送系统的准确性和效率，提升系统的实用性，未来工作主要有：扩充科技政策语料库，通过增大数据量提高个性化推送的多样性、准确度；优化NLP模型，提高模型准确性；进一步完善小程序的功能，优化界面的使用。