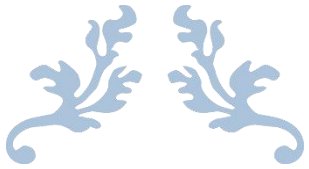

# 电子卷宗检索技术方案

2019-7-16

南京大学软件学院
李行行

目录

# 1 引言

## 1.1 写作目的

本文档是对电子卷宗检索系统的详细描述，包括项目背景，目标客户，功能实现，以及系统建构和使用到的技术。从而使用户，项目参与人员能对系统的主要功能、实现方式具有一致的理解。

## 1.2 项目背景

目前，在法院系统中，很多文件以图片的形式进行存储，包括拍照取证的照片和扫描件等，这就为搜索增加了难度，不能够快速精确的对图片中的文字进行搜索。随着深度学习的发展，文字识别技术以及相当成熟，使用 OCR 技术对图片进行文字的识别和提取，并将识别的结果以 json 的格式存储并导入 ElasticSearch 中，当数据量大时，为了持久化存储，也可将数据导入 mongodb 中，使用 ElasticSearch 能够对大量数据进行搜索和分析，从而实现对图片文字进行搜索的功能。

本系统初步目标客户为法院工作人员，也可将该系统运用于其他需要对图片文字进行搜索的领域。

# 2 系统总体设计

## 2.1 系统功能架构

本系统主要提供对数据的搜索功能，参照主流搜索引擎设计，主要分为三个功能页面，分别为搜索页面，搜索结果页面，结果详情页面。

1. 在搜索页面的搜索框内填写需要搜索的内容，会跳转到搜索结果页面。
2. 搜索结果页面根据相关度进行排序，每个页面展示十条数据，有分页功能，对搜索内容进行标红处理，页面上方也有搜索框，可以继续进行搜索，页面左侧为友情链接，右侧为搜索历史，可以直接点击进行再次搜索。
3. 点击每一条结果可以跳转到详情界面，详情界面左边为原始图片，右边为图片对应的内容。并对匹配到的内容进行标红处理。

## 2.2 系统技术架构

### 2.2.1 Django(python web 框架)

Django 是一个开放源代码的 Web 应用框架，由 Python 写成。Django 是一个基于 MVC 构造的框架。但是在 Django 中，控制器接受用户输入的部分由框架自行处理，所以 Django 里更关注的是模型（Model）、模板(Template)和视图（Views），称为 MTV 模式。它们各自的职责如下：

| 层次 | 职责 |
| --- | --- |
| 模型（Model），即数据存取层 | 处理与数据相关的所有事务： 如何存取、如何验证有效性、包含哪些行为以及数据之间的关系等。 |
| 模板(Template)，即表现层 | 处理与表现相关的决定： 如何在页面或其他类型文档中进行显示。 |
| 视图（View），即业务逻辑层 | 存取模型及调取恰当模板的相关逻辑。模型与模板的桥梁。 |

从以上表述可以看出 Django 视图不处理用户输入，而仅仅决定要展现哪些数据给用户，而 Django 模板 仅仅决定如何展现 Django 视图指定的数据。或者说，Django 将 MVC 中的视图进一步分解为 Django 视图 和 Django 模板两个部分，分别决定 “展现哪些数据” 和 “如何展现”，使得 Django 的模板可以根据需要随时替换，而不仅仅限制于内置的模板。至于 MVC 控制器部分，由 Django 框架的 URLconf 来实现。URLconf 机制是使用正则表达式匹配 URL，然后调用合适的 Python 函数。URLconf 对于 URL 的规则没有任何限制，你完全可以设计成任意的 URL 风格，不管是传统的，RESTful 的，或者是另类的。框架把控制层给封装了，无非与数据交互这层都是数据库表的读,写,删除,更新的操作。在写程序的时候，只要调用相应的方法就行了，感觉很方便。程序员把控制层的东西交给 Django 自动完成了。 只需要编写非常少的代码完成很多的事情。所以，它比 MVC 框架考虑的问题要深一步，因为我们程序员大都在写控制层的程序。现在这个工作交给了框架，仅需写很少的调用代码，大大提高了工作效率。

Django 的主要目的是简便、快速的开发数据库驱动的网站。它强调代码复用，

多个组件可以很方便的以“插件”形式服务于整个框架，Django 有许多功能强大的第三方插件，你甚至可以很方便的开发出自己的工具包。这使得 Django 具有很强的可扩展性。它还强调快速开发和 DRY(Do Not Repeat Yourself)原则。

Django 基于 MVC 的设计十分优美：

1.对象关系映射 (ORM,object-relational mapping)：以 Python 类形式定义你的数据模型，ORM 将模型与关系数据库连接起来，你将得到一个非常容易使用的数据库 API，同时你也可以在 Django 中使用原始的 SQL 语句。
URL 分派：使用正则表达式匹配 URL，你可以设计任意的 URL，没有框架的特定限定。像你喜欢的一样灵活。

2.模版系统：使用 Django 强大而可扩展的模板语言，可以分隔设计、内容和 Python 代码。并且具有可继承性。

3.表单处理：你可以方便的生成各种表单模型，实现表单的有效性检验。可以方便的从你定义的模型实例生成相应的表单。

4.Cache 系统：可以挂在内存缓冲或其它的框架实现超级缓冲 —— 实现你所需要的粒度。

5.会话(session)，用户登录与权限检查，快速开发用户会话功能。

6.国际化：内置国际化系统，方便开发出多种语言的网站。

7.自动化的管理界面：不需要你花大量的工作来创建人员管理和更新内容。Django 自带一个 ADMIN site,类似于内容管理系统。

Django 工作机制：

1.用 manage.py runserver 启动 Django 服务器时就载入了在同一目录下的 settings.py。该文件包含了项目中的配置信息，如前面讲的 URLConf 等，其中最重要的配置就是 ROOT_URLCONF，它告诉 Django 哪个 Python 模块应该用作本站的 URLConf，默认的是 urls.py

2.当访问 url 的时候，Django 会根据 ROOT_URLCONF 的设置来装载 URLConf。

3.然后按顺序逐个匹配 URLConf 里的 URLpatterns。如果找到则会调用相关联的视图函数，并把 HttpRequest 对象作为第一个参数(通常是 request)。

4.最后该 view 函数负责返回一个 HttpResponse 对象。

### 2.2.2 ElasticSearch：

ElasticSearch 是一个分布式、高扩展、高实时的搜索与数据分析引擎。它能很方便的使大量数据具有搜索、分析和探索的能力。充分利用 ElasticSearch 的水平伸缩性，能使数据在生产环境变得更有价值。ElasticSearch 的实现原理主要分为以下几个步骤，首先用户将数据提交到 Elastic Search 数据库中，再通过分词控制器去将对应的语句分词，将其权重和分词结果一并存入数据，当用户搜索数据时候，再根据权重将结果排名，打分，再将返回结果呈现给用户。Elasticsearch 是与名为 Logstash 的数据收集和日志解析引擎以及名为 Kibana 的分析和可视化平台一起开发的。这三个产品被设计成一个集成解决方案，称为“Elastic Stack”（以前称为“ELK stack”）。

Elasticsearch 可以用于搜索各种文档。它提供可扩展的搜索，具有接近实时的搜索，并支持多租户。”Elasticsearch 是分布式的，这意味着索引可以被分成分片，每个分片可以有 0 个或多个副本。每个节点托管一个或多个分片，并充当协调器将操作委托给正确的分片。再平衡和路由是自动完成的。“相关数据通常存储在同一个索引中，该索引由一个或多个主分片和零个或多个复制分片组成。一旦创建了索引，就不能更改主分片的数量。

Elasticsearch 使用 Lucene，并试图通过 JSON 和 Java API 提供其所有特性。它支持 facetting 和 percolating，如果新文档与注册查询匹配，这对于通知非常有用。另一个特性称为“网关”，处理索引的长期持久性；例如，在服务器崩溃的情况下，可以从网关恢复索引。Elasticsearch 支持实时 GET 请求，适合作为 NoSQL 数据存储，但缺少分布式事务。