实验报告

彭培烜 2018202201

1. **问题背景**

arXiv是用于搜索，浏览和下载学术论文的重要工具，为公众和研究社区提供了开放获取学术论文的服务，涉足物理学和计算机科学的众多领域。为了让arXiv可用度更高，康奈尔大学在kaggle上创建了一个免费、开放的数据集，包含了170多万学术论文。

此次任务的目标，是基于kaggle上的arXiv数据集，建立一个AI应用程序，能够进行问题搜索，建立知识图谱，进行文章推荐等功能。当前阶段已经完成了问答系统的实现。

1. **问答系统的实现**
2. **概述**

该项目基于kaggle上提供的数据集进行操作，该数据初始为json文件，又以下特征：

id: ArXiv ID

submitter: 提供者

authors: 作者

title: 标题

comments: 其他信息，如页数和数字

journal-ref: 发表该论文的杂志和信息

doi

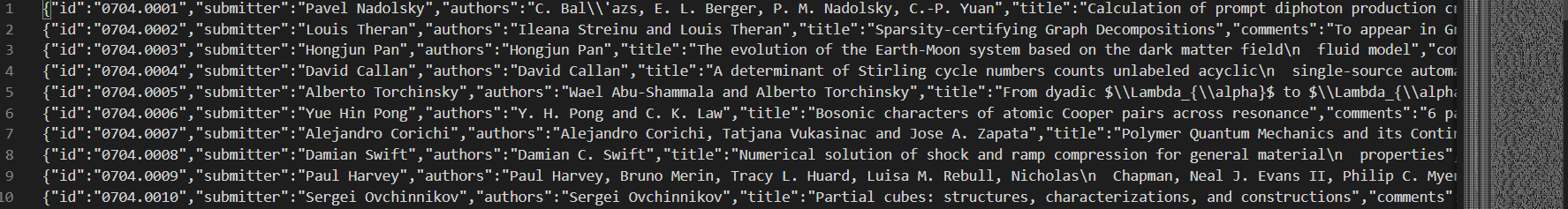
abstract: 摘要，此次检索系统便基于该属性进行回答

categories: ArXiv系统中的类别或标记

version: 版本

因为论文的全文是在谷歌云中的PDF当中，所占空间过大，而且PDF难以处理，因此该问答系统从每篇文章的摘要中进行搜索。

**数据截图：**

****

1. **Haystack**

本次实验使用Haystack处理如此大量的数据，它允许将Q&A模型扩展到大型文档集合。具体信息在<https://github.com/deepset-ai/haystack>。

Haystack有三个主要组成部分：

**Document Store：**用于存储搜索文档的数据库，其主要推荐的是ElasticSearch。

**Retriever：**从大量文档中识别候选段落的快速，简单的算法，包括TF-IDF和BM25等，有助于将Reader的范围缩小到可以回答给定问题的较小文本单元。

**Reader：**神经模型，用于从文本中寻找答案。

**安装Haystack：**

通过GitHub安装：

pip install git+https://github.com/deepset-ai/haystack.git

1. **设置文档库**

Haystack通过存储在文档库中的文档查询答案，目前的文档库有ElasticsearchDocumentStore，SQLDocumentStore和InMemoryDocumentStore。这里我们使用的是ElasticserchDocumentStore，因为它预装了全文查询，BM25检索和文本嵌入向量储存等功能。

**安装ElasticserchDocumentStore：**

wget https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.6.2-linux-x86\_64.tar.gz -q

tar -xzf elasticsearch-7.6.2-linux-x86\_64.tar.gz

chown -R daemon:daemon elasticsearch-7.6.2

安装和配置完成后，需要将数据导入。

导入ElasticserchDocumentStore需要一个字典列表。此次实验中，我们使用title作为名称，abstract作为文本传递。

1. **配置Retriever，Reader和Finder**

**Retriever：**

这里使用默认的BM25算法

retriever = ElasticsearchRetriever(document\_store=document\_store)

**Reader：**

这里使用FARM中的RoBERTa QA模型

reader = FARMReader(model\_name\_or\_path="deepset/roberta-base-squad2", use\_gpu=True, context\_window\_size=500)

**Finder：**

finder = Finder(reader, retriever)

至此，一个问答系统已经设计完成

1. **下一步计划**

在第一阶段的实现过程中，在配置环境上遇到了一些问题，导致进度稍慢，但也解决了环境配置和数据预处理中的一些困难。下一步我们计划实现标题预测以及论文推荐。