

数据科学基础--中期报告

小组信息: 共 2 人

191250177 杨骏丰 模型训练和 word2vec 部分代码

181250014 陈文龙 bert 算法和联合算法整合

邮箱: 1011921795@qq.com

研究问题: 文本匹配

研究背景: 智慧政务服务综合平台由“政务服务门户”、“政务服务管理系统”、“政融选择系统”、“事项目录管理系统”、“智能决策支持系统”、“数据共享系统”、“电子监察监管系统”、“统一受理平台”等系统作为基础支撑,覆盖线上线下多种渠道,实现服务场景多样、数据交换共享、科技水平领先的智慧服务,涵盖社保/医保,教育就业,婚姻生育等等各项民生领域,为公众提供方便快捷、公平普惠、优质高效的网上政务服务。是各级政府加快政府职能转变,持续深化“放管服”改革的重要抓手。每一项服务背后都依托于不同的数据支撑,但由于不同地区、不同政府部门对于数据的标准不同,也就导致了同一数据有不同的表现形式,例如在公安部门内数据表现为“姓名”,在民政局内的数据表现为“名称”,都表示为同一个意思,以往的工作是通过人工的手段将这些同义字段关联到事项目录管理系统中的标准字段,比如将“姓名”,“名称”都统一成“姓名”这个标准数据元,但这种方法耗时耗力,而且往往关联某个标准数据元时需要依靠外部的辅助信息,例如下列的“设计利用储量”关联到标准数据元字段“矿产设计利用储量”就需要利用额外的“自然资源”这个字段作为判断,即我们的场景任务并不简单的是一对一的匹配,有时候需要涉及到多对一的匹配。

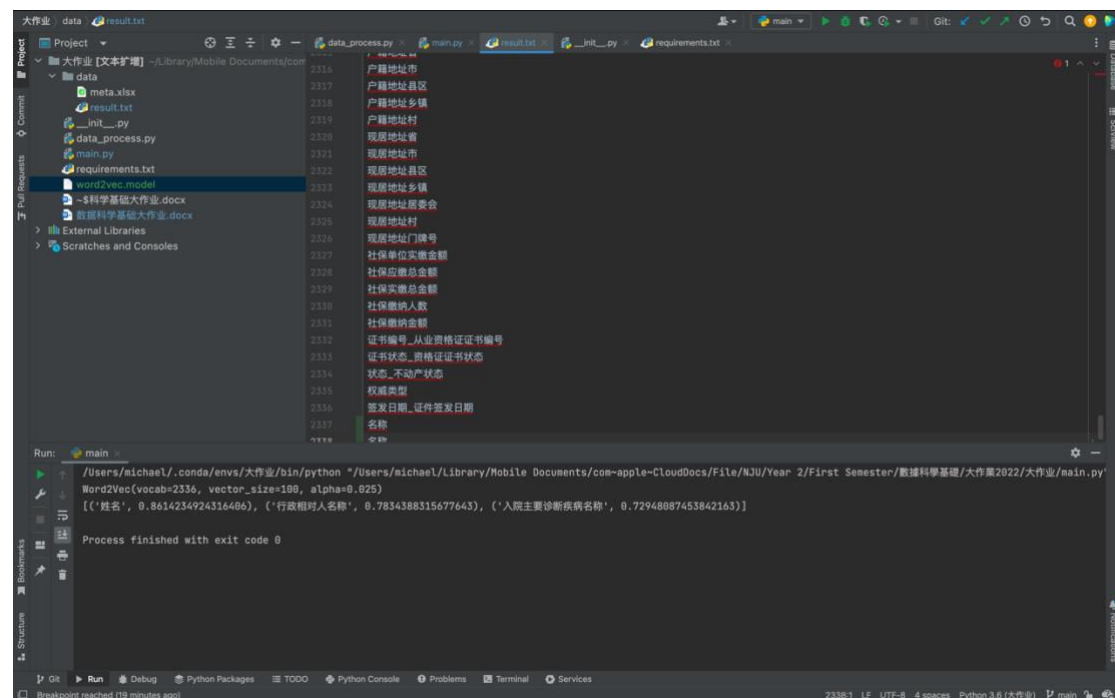
研究方法: 从数据源获取数据后,先对数据去重,再使用基于词向量的 word2vec 对数据集进行训练,使用训练好的模型进行文本的相似度计算,输入目标匹配字段后,对训练后的数据集算出与目标字段的「距离」后,得出与目标字段最相近即相似度最高的 TOP k 个字段。由于字段关联关系,存在一些字些是需要利用额外的辅助信息去进行关联,为了更好地实现同义字段的一对一的关联匹配,此处我们需要引入 BERT 模型去对数据进行处理。BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018) 是一种预训练语言表示的方法。但与原始 transformers (Vaswani et al., 2017) 的主要区别是, BERT 没有解码器,但在基本版本中堆叠了 12 个编码器,而且在更大的

预训练模型中会增加编码器的数量。这种架构不同于 OpenAI 的 GPT-2，它是适合自然语言生成（NLG）的自回归语言模型。

案例分析： 例如我们使用语料库，读入数据后进行去重，并通过预训练得到一个可用的模型，简单测试模型，验证确实可用。我们最后输入”姓名”，返回值是”信访人姓名”及其相似度，但是部分字段间有通过外部信息的关联，结合 BERT 训练出一个可以识别出这种字段的模型，联合两种方法，可以得到一个更符合要求的程序。

附加点： 对于动态加载，我们想到的方法是定时检查数据源，如果发生修改，会马上把新内容加到训练模型里，并即时运行结果。对于多对一的匹配，我们的方法是把输入的多个字段内容都进行相似度计算，结合计算他们的相似度，将最优结构输出。

当前程序截图：



The screenshot shows a Python IDE with a project named '大作业'. The project structure includes files like 'data', 'meta.xlsx', 'result.txt', 'init.py', 'data_process.py', 'main.py', 'requirements.txt', and 'word2vec_model'. The 'main.py' file is open, displaying a list of fields and their corresponding similarity scores. The fields are listed in a table-like format with two columns: the field name and the similarity score. The fields include '户籍地址市', '户籍地址县区', '户籍地址乡镇', '户籍地址村', '现居地址省', '现居地址市', '现居地址县区', '现居地址乡镇', '现居地址居委会', '现居地址村', '现居地址门牌号', '社保单位实缴金额', '社保单位应缴金额', '社保单位缴费基数', '社保缴费基数', '证书编号_从业资格证书编号', '证书状态_资格证书状态', '状态_不动产状态', '权威类型', '签发日期_证件签发日期', and '名称'. The similarity scores are numerical values ranging from 0.72948887453842163 to 0.8614234924316486. The bottom of the IDE shows the output of the program, which is a list of tuples containing the field name and its similarity score.

```
Word2Vec(vocab=2336, vector_size=100, alpha=0.025)
[('姓名', 0.8614234924316486), ('行政相对人名称', 0.7834388315677643), ('入院主要诊断疾病名称', 0.72948887453842163)]
```

Process finished with exit code 0