**泰迪内推平台招聘与求职双向推荐系统构建**

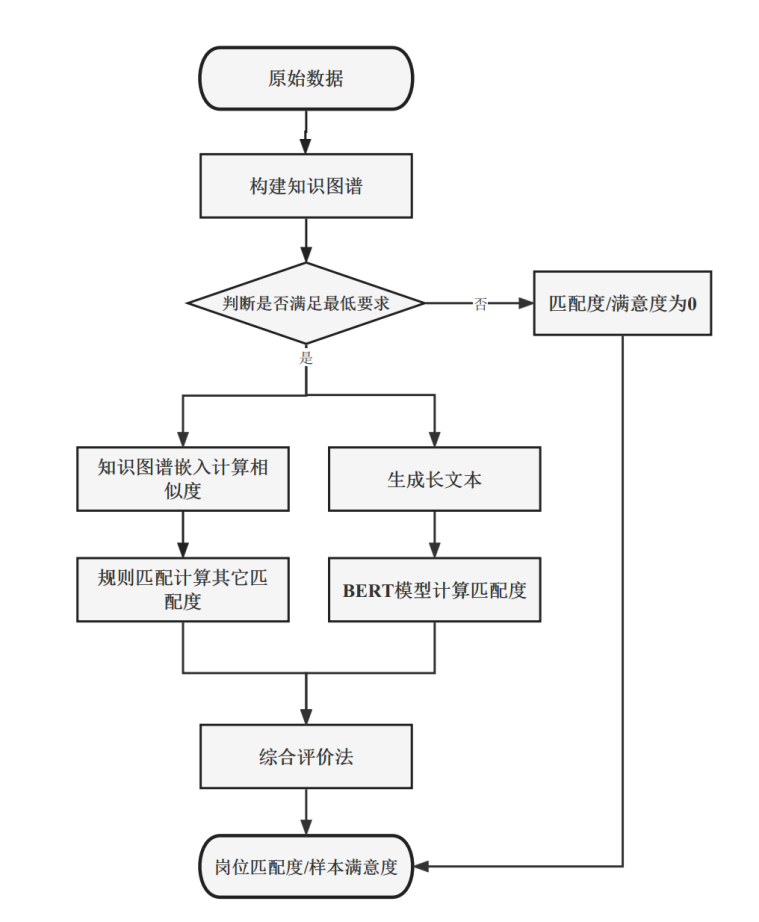
# 五、问题三：基于知识图谱和BERT的求职者匹配和满意模型

## 5.1问题三分析

问题三主要是针对企业选取求职者与求职者选择企业时所产生的匹配度和满意度进行建模计算。本题我们认为匹配程度是依据求职者对企业要求的满足程度，即求职者是否满足企业最低要求以及满足各项要求的多少，而求职者的满意程度是依据求职者对企业薪资待遇、福利状况、工作地点等是否达到求职者的最低要求以及达到了多少。根据查找文献，求职者匹配度和满意度在一般要求方面是一致的，而在最低要求与福利待遇方面是需要单独讨论的。由于本题所提供的数据集仅包含岗位数据和求职者数据，并不包含岗位和求职者的匹配数据，因此本题关于岗位匹配度和求职者满意度的计算问题实际上是一道无监督的相似性计算问题。

为此，我们从规则匹配和语义匹配两部分出发，首先对求职者的各类经历和企业的工作要求进行关键字提取，并根据求职者和企业的属性特征建立知识图谱模型，通过设置企业最低要求和求职者最低要求，过滤不符合企业最低要求的求职者和不符合求职者最低要求的企业，从而根据知识图谱的节点相似度来计算企业与求职者一般匹配度，并根据建立自定义规则，来分别计算匹配度与满意度。同时，我们对求职者和企业的各类属性进行长文本合并，并输入BERT模型中进行编码，通过计算余弦相似度得到求职者和企业的语义相似度，最后根据综合评价法，得到岗位匹配度和求职者满意度。

下图是本节的框架：



## 5.2基础理论

### 5.2.1知识图谱

知识图谱（knowledge graph）是一种基于语义网络的知识表示方法，它将实体与实体之间的关系通过节点和边的方式进行建模，构造一个结构化的动态数据库，从而方便人类对所存储的数据集进行查看、理解与应用，知识图谱目前广泛应用于金融领域、推荐领域、教育领域等。知识图谱的建立往往需要借助深度学习中自然语言处理技术、信息抽取技术，借助深度学习技术，可以大幅度理解文本语义，获取更精准的实体信息，从而打造一个精准、可信的图数据库。

知识图谱的核心就是实体、属性、关系。实体是指现实中具体存在的事物，包括人、事件、组织、概念等。属性是指实体的属性，可以是实体的名称、类别、描述等。关系是实体与实体之间的关系，用来串联起实体与实体之间的关系。

知识图谱的建立通常包括以下步骤：

1. 数据集的采集：从数据源中采集结构化数据、半结构化数据与非结构化数据。
2. 数据预处理：对采集到的数据集进行清洗、删除停用词、分词等。
3. 信息抽取：对长文本中的信息进行实体抽取、关系抽取、属性抽取，提取出有用的信息。
4. 知识融合：对数据集中存在歧义的实体进行处理。
5. 知识检验：对存储的数据集进行检验，判断是否存入了有效的数据。
6. 知识推理与应用：利用最终得到的图数据库应用到别的下有任务重，比如智能问答、智能推荐、人岗匹配等。

### 5.2.2 BERT模型

BERT（Bidirectional Encoder Representations from Transformers）是一种预训练语言模型，由Google在2018年提出。与传统的单向语言模型不同，BERT是一种双向的语言模型，可以同时利用上下文信息和自身信息来预测一个词的可能性，从而提高了语言模型的效果。BERT的出现，对于自然语言处理领域的发展产生了重要的影响。

BERT的模型结构主要由两部分组成，分别是Transformer编码器和任务特定的输出层。在预训练阶段，BERT使用了两种预训练任务来训练模型，分别是Masked Language Model（MLM）和Next Sentence Prediction（NSP）。

MLM任务是在输入序列中随机选择一些词进行遮盖，然后让模型预测这些被遮盖的词。这种任务可以让模型学习上下文信息，同时也能够让模型学习到词与词之间的关系。

NSP任务是判断两个句子是否是相邻的句子。通过让模型预测两个句子是否相邻，可以让模型学习到句子之间的语义关系，从而提高模型的表现。

在微调阶段，BERT可以应用到多种自然语言处理任务中，如文本分类、命名实体识别、关系抽取、机器翻译等。通过微调，BERT可以利用任务特定的标注数据来调整模型参数，从而提高模型在该任务上的表现。

总的来说，BERT是一种非常重要的自然语言处理模型，它利用了双向语言模型和Transformer模型的优势，通过大规模的无监督学习来学习语言的表示，从而提高了各种自然语言处理任务的效果。BERT的出现，对于自然语言处理领域的发展产生了重要的影响，未来也会继续发挥重要作用。

### 5.2.3综合评价法

## 5.3 规则匹配度

在本文中，规则匹配度是根据题意、相关文献参考与专家意见对岗位匹配度和求职者满意度的匹配度制定自定义规则，从而在规则下进行计算。规则主要包括两个方面：最低要求与特殊要求。根据题意，求职者不满足岗位最低要求的匹配度为0，同理，岗位不满足求职者最低要求的满意度为0，接下来从求职者和岗位两方面来分别介绍岗位匹配度与求职者满意度。从数据来看，岗位与求职者的最低要求基本一致，从工作性质、薪酬、工作地点、工作经验、教育水平五个方面进行计算。

|  |  |  |
| --- | --- | --- |
| 要求 | 要求范围 | 最低要求 |
| 工作性质 | 全职、兼职 | 和岗位一致 |
| 薪酬 | 最低薪资、最高薪资 | 求职者最低薪资<岗位最高薪资 |
| 工作地点 | 省、市、区 | 求职者期望的省=岗位所在省 |
| 工作经验 | 数值型 | 求职者工作经验>岗位要求最低工作经验 |
| 教育水平 | 不限、技工、大专、本科、硕士、博士 | 求职者教育水平=岗位要求教育水平 |

### 5.3.1 岗位匹配度

根据文献查询，我们将岗位匹配度分为以下几个部分。

薪资匹配度：计算求职者的薪资与岗位薪资的重合率。即通过（求职者薪资与岗位薪资的交集）/（求职者薪资与岗位薪资的并集），重合率范围在（0，1）之间，重合率越高，匹配度也越高。

工作经验匹配度：计算求职者工作经验相对岗位要求工作经验的超出率。通过计算（求职者工作经验-岗位要求工作经验），并分岗位进行归一化处理，将范围放缩到（0，1）之间。

教育水平匹配度：计算求职者教育水平相对岗位要求教育水平的超出率。通过计算（求职者教育水平数值编码-教育水平数值编码），同样进行归一化。

工作地点匹配度：计算求职者工作地点与岗位工作地点的重合率，如果省不一致，则匹配度为0；如果省份匹配上，则匹配度为1/3，如果市也匹配上，则匹配度为2/3，如果区也匹配上，则匹配度为1。

技能匹配度：计算求职者技能与岗位要求的语义匹配度，通过BERT模型进行计算。

### 5.3.2 求职者满意度

根据文献查询，我们将求职者满意度分为以下几个部分：

薪资满意度：计算岗位所提供的最高薪资与求职者最高要求薪资的差值，因为存在正负值，因此同样需要进行归一化。

福利待遇满意度：我们将福利待遇进行归纳，包括以下福利。专业培训、五险一金、交通补贴、免费住宿、免费班车、出国机会、员工旅游、定期体检、年终奖金、弹性工作、绩效奖金、股票期权、通讯补贴、餐饮补贴，其中我们认为五险一金是最重要的，如果福利待遇中没有五险一金，则认为满意度为0

公司满意度：在本文所搜集到的数据集中，主要针对公司的公司性质、公司规模、公司注册资本来计算满意度，根据市场调查结果显示，公司性质对人才的吸引力从大到小排序分别为：外资、外企、上市公司、合资、国企、民营企业，公司规模与公司资产越大，对人才的吸引力也越大。

### 5.3.3 综合评价

## 5.4 供求职知识图谱模型的建立

本节将根据数据预处理后的数据集构建供求职知识图谱，从而进行图谱可视化展示与点相似度计算。由于本节使用的数据集属于招聘领域数据集，具有专业性高、技术类与领域类专有名词多的特点，比如技术类名称：Python、Java、R等，领域类名称：教育领域、金融领域、旅游领域、推荐领域等，而在4.2节中，我们已经对该数据集进行了高精度的数据标注，可以进行接下来供求职知识图谱的构建。经过知识图谱的构建，将其存储到Neo4j图数据库中，并根据图数据科学库来计算求职者节点和岗位节点的相似度。

### 5.4.1 供求职知识图谱

供求职知识图谱，是将求职者简历与公司岗位的知识提取后，形成三元组结构，从而构建一个供求职关系网络。在这两个数据集中，每一个节点代表一个短字段的值，每一个边代表ID与点之间的特征列，以ID节点为中心，连接其它节点，形成每一位求职者的身份信息网络和每一个岗位的岗位信息网络，由于节点之间是相连的，不存在一个节点只属于一个ID的情况，而是多个ID共用同一个节点，因此我们可以通过求职者ID的信息网络与岗位ID的信息网络节点取交集来计算点的相似度，即求职者与岗位的匹配度。

在求职者数据集中，我们选取以下特征输入知识图谱：

|  |  |
| --- | --- |
| 字段类型 | 特征 |
| 短字段 | ID, 用户名, 地址, 省, 市, 区, 性别, 求职状态, 上岗时间, 生日, 经验, 政治面貌, 工作性质, 最低薪资, 最高薪资, 工作时间, 项目时间, 训练时间, 教育时间, 应届身份 |
| 分词：短字段 | 岗位, 行业, 公司, 行业, 岗位, 公司, 项目名称, 项目人员, 公司, 培训项目, 学历, 学校, 专业 |
| 分词：技能 | 竞赛情况, 证书情况, 语言情况, 技能, 关键词情况 |
| 分词：长字段 | 自我评价关键词, 工作描述关键词, 项目描述关键词, 项目成就关键词, 培训描述关键词 |

由于部分短字段存在多个值，因此在知识图谱中创建节点的时候需要单独进行分词再进行创建，技能和长字段同理，从而保证知识图谱中各节点内容的单一性。岗位数据集的知识图谱同样也一样进行处理。

在岗位数据集中，我们选取以下特征输入知识图谱

|  |  |
| --- | --- |
| 字段类型 | 特征 |
| 短字段 | ID, 公司, 岗位, 工作性质, 最低薪资, 最高薪资, 支付方式, 经验, 学历, 人数, 福利, 地址, 地理编码-省, 地理编码-市, 地理编码-区, 地址, 省, 市, 区, 截止时间,缩写, 公司性质, 人数 , 资本, 邮箱, 电话 |
| 分词：短字段 | 职责, 关键词, 技能要求, 行业 |
| 分词：长字段 | 关键词 |

### 5.4.2知识图谱的构建框架

知识图谱的构建一般包括以下几个步骤：本体设计、信息抽取、数据标注、知识融合、知识存储与可视化展示。在本文中，由于在4.2节中就已经通过数据标注进行了分词处理，因此本小节不需要进行额外的数据标注。

1. 本体设计

本体是关于某个主题的概念性和说明性表示，包括领域、各类型对象的名称、定义及相关描述关系。通过确定本体，可以进一步确定知识图谱的实体的构成、概念和约束关系，并根据本体进行实例化，最终生成的知识图谱依据于本体的设计的框架之下。本体的设计步骤包括确定领域、定义类和层次结构、定义属性和关系、定义约束和关系、举例和创建实例。本体设计的准则包括明确性和客观性、一致性、可扩展性和位于知识这一层次。在本文中，我们将通过知识驱动的方式，生成求职者简历与岗位要求的实体架构。

1. 信息抽取

信息抽取是知识图谱构建中十分重要的一步，是针对结构化数据与非结构化数据信息提取的过程，主要包括实体抽取、关系抽取与属性抽取三个部分。实体抽取也称命名实体识别(NER)，从非结构化数据中该自动提取专有名词，在本文中，主要是基于jieba分词的方式对关键字进行提取，关系抽取是从实体的上下文中挖掘实体之间的关系，从而将实体连接起来，本文所提供的数据已经提供了关系列，无需进行关系抽取。属性抽取与关系抽取类似，从实体的上下文或外部提取该实体的属性，例如在工作岗位缺失的数据上，可以通过百度百科、维基百科等方式挖掘更多的属性关系。

信息抽取的模型根据不同的数据集或任务，有基于词频方法的分词模型，通过TF-IDF、TextRank等抽取关键词；基于深度学习的信息抽取模型，主流的模型有BiLSTM、CRF模型；基于大型语言模型的信息抽取模型，通过有BERT模型对实体进行抽取。

本文主要采取基于词频的方法对关键词进行提取，并配合人工数据标注的方式，在增加数据质量的同时大幅度减少时间成本。

1. 数据标注

数据标注对与构建个性化知识图谱是十分关键的，并且需要大量的时间成本，采用特定的格式对文本进行词性标注，包括人工标注和自动标注两种方式，自动标注可以使用Jieba、Hanlp等模型进行标注。本文同时采用这两种方式，首先通过jieba对文本进行分词，再进行知识检验与人工标注，构建自定义词典之后重新进行二次分词，最终得到的关键字结果相较于第一次的自动标注有显著的提升。

1. 知识融合

在获得知识后，需要进行知识融合来消解歧义和矛盾，其中最重要的是实体对齐，有些实体具有不同的含义，比如：“语言”在编程中主要是指编程类的语言，在人类社会中是指交流的语言，需要进行区分，同时，有些实体是指向现实社会中的同一事物，比如大小写没有区别的英文单词：“Mysql”与“MySql”等，需要进行融合。

在本文中，主要是通过计算Levenshtein距离来判断两个实体之间距离的大小，将Levenshtein距离较小的合并为一个节点，可以有效避免实体之间由于大小写而产生的差异。

1. 知识存储与可视化展示

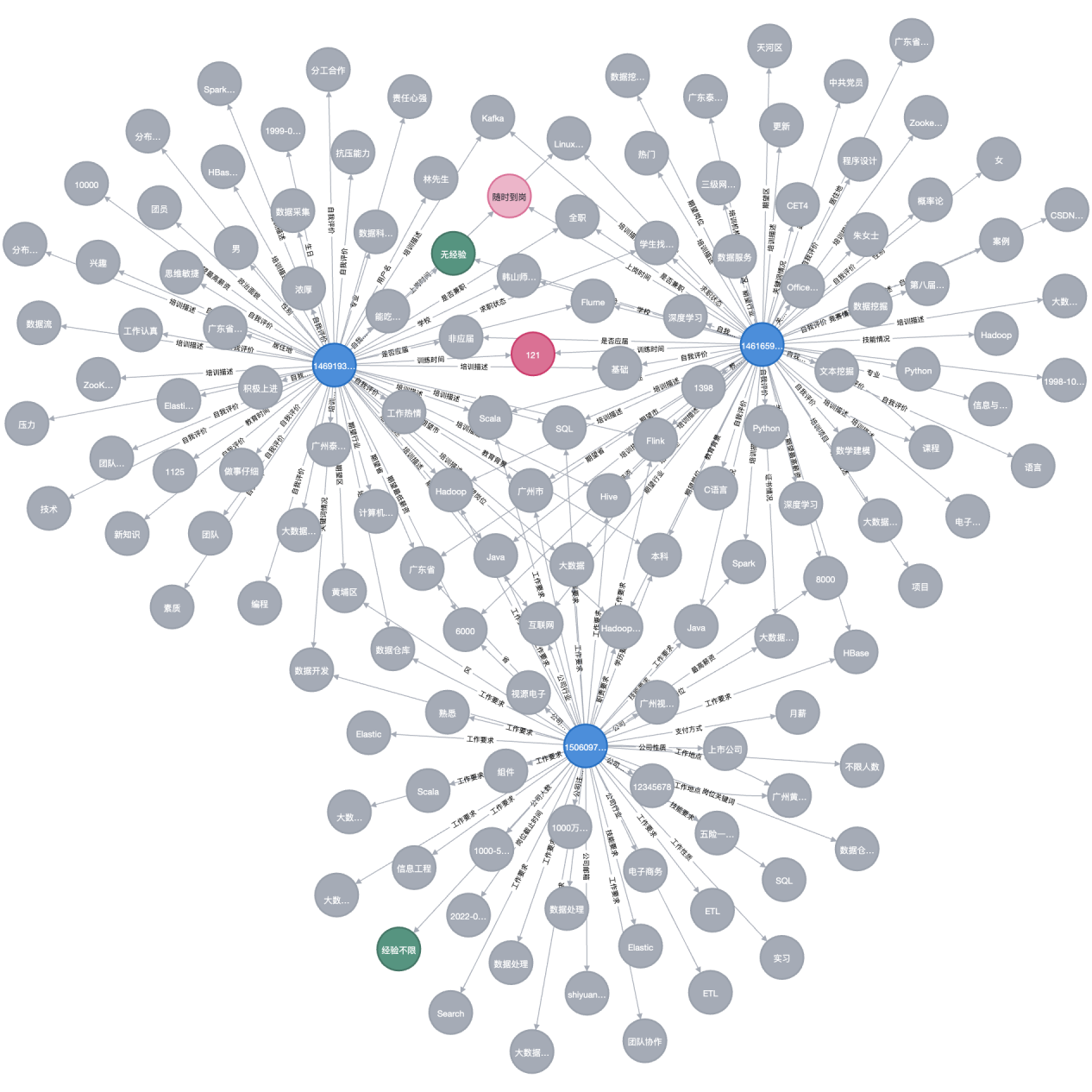
知识图谱的数据存储一般有两种方式，一种是基于RDF格式进行存储，另一种是基于图数据库的存储，而图数据库在在图数据库上，图数据库是一种特殊的数据库类型，它的数据模型是基于图论的。图数据库的数据结构是由节点和边构成的图形结构，节点和边都可以拥有属性，这些属性可以用于描述节点和边之间的关系。与传统的关系型数据库不同，图数据库在处理复杂的数据关系时表现得更加灵活和高效，因为它可以轻松地进行多级关系的查询和分析。图数据库通常被用于处理大规模的数据集，尤其是在社交网络、推荐系统、反欺诈、生物信息学等领域中具有广泛的应用。目前比较主流的图数据库有：星云、neo4j。本文将数据集存储在neo4j图数据库上。

### 5.4.3 neo4j知识存储与可视化

Neo4j图数据库是一种将数据转化为可动态拉动的关系网络的图数据库，将数据分布在网状结构中，以强大和稳定的性能出名。在Neo4j的一张知识图谱中，里面包含点和边，不同的点、边都可以通过不同的颜色进行标注，在一张知识图谱中，可以根据节点、边进行复杂的检索行为，Neo4j数据库使用的是Cypher语言进行操作，包含对节点、边的增删改查等一系列操作，在Python中可以使用Py2neo模块对数据库进行连接，实现在python上对数据库进行一系列的操作。

本文将求职者数据集与岗位数据集抽取部分特征导入Neo4j数据库中，以下为数据集导入图数据库的操作流程：

1. 连接Neo4j数据库。使用“graph = Graph('http://localhost:7474/',auth = (username,password))”进行连接。
2. 创建节点。首先使用“node = NodeMatcher(graph).match(name=str(data)).first()”检查是否存在已经创建了同名称的节点，如果不存在节点，则使用“node = Node(label, name = name)”与“graph.create(node\_sigleword)”来创建节点，如果存在节点，其次判断已存在节点的标签是否也一样，如果一样，则跳过创建节点的环节，如果不一样，则使用“node.add\_label(label)”与“ graph.push(new\_label)”来为已存在的节点增加新的标签。
3. 创建边。使用“node\_id = NodeMatcher(graph).match(name=str(id)).first()”找到对应的ID，从而使用“relation = Relationship(node\_id,column\_relation,node)”与“graph.create(relation)”来创建边，从而为node\_id与node创建一条边。查询节点及其关系。这里以ID为“1461659304984707072”和ID为“1469193231341322240”的求职者与岗位ID为“1506097472638943244”的岗位为例，使用MATCH (a1:ID {name: '1461659304984707072'})-->(b1) MATCH (a2:ID {name: '1506097472638943244'})-->(b2) MATCH (a3:ID {name: '1469193231341322240'})-->(b3) return a1,b1,a2,b2,a3,b3”来进行查询，查询结果如下：



在知识图谱中，理论上我们可以为每个节点都标上颜色，但是实用性不大，且需要耗费大量的时间成本和人力成本，因此这里仅用来作为数据的展示。

### 5.4.4 节点相似度模型

节点相似度的计算是基于 Jaccard 或 Overlap 相似性度量，在Neo4j图数据库中，如果两个节点共享多个邻居节点，则认为这两个节点是相似的，因此可以根据共享节点的重叠个数来计算节点相似度。而本文使用的是Jaccard相似度分数，它的计算公式如下：

/private/var/folders/c3/vyxk7z2n5dx_dgz2fc4gsz440000gn/T/com.kingsoft.wpsoffice.mac/wpsoffice.aCxsTVwpsoffice

该算法的输入是两个节点及其各自的邻居节点，统计从第一个节点出发到达第二个节点的个数以及两个节点领居节点的个数，从而来计算相似度，这样的好处是在Neo4j上能清除看到节点之间的相似度，从直观上来看，就是领居节点个数越多，则相似度越高。

在Neo4j图数据库的图数据科学库（GDS）中，我们可以调用GDS的API来计算Neo4j中存储的两个节点的相似度，这里以ID为“1461512488951611392”的求职者为例，使用代码“MATCH (a1:ID {{name: '{resume\_id}'}})-->(b1) WITH a1, collect(id(b1)) AS p1 MATCH (a2:ID {{name: '{joblist\_id}'}})-->(b2) WITH a1, p1, a2, collect(id(b2)) AS p2 RETURN a1.name AS from, a2.name AS to, gds.similarity.jaccard(p1, p2) AS similarity”，来计算相似度，并对相似度以求职者ID或招聘信息ID为分组进行分组归一化，得到结果如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 求职者 ID | 招聘信息 ID | 匹配度 | 归一化匹配度 |
| 1496340730497597440 | 1479027851352211456 | 0.0494 | 0.1670 |
| 1496340730497597440 | 1479026368565084160 | 0.0610 | 1.0000 |
| 1496340730497597440 | 1479024040038498304 | 0.0482 | 0.0815 |
| 1496340730497597440 | 1479021937257742336 | 0.0476 | 0.0403 |
| 1496340730497597440 | 1479020791432609792 | 0.0482 | 0.0815 |
| 1496340730497597440 | 1466305532775235584 | 0.0471 | 0.0005 |
| 1496340730497597440 | 1466304423818035200 | 0.0471 | 0.0005 |
| 1496340730497597440 | 1466303717568544768 | 0.0476 | 0.0403 |
| 1496340730497597440 | 1466303123046924288 | 0.0471 | 0.0005 |
| 1496340730497597440 | 1466295233930199040 | 0.0602 | 0.9472 |
| 1496340730497597440 | 1466294662284312576 | 0.0471 | 0.0005 |

从相似度值，可以明显发现，知识图谱所计算的相似度较小，通过Neo4j的查询语句，发现求职者和岗位之间所共享的节点往往较少，这是由于求职者和岗位往往很难出现重叠的节点，而岗位本身的节点一般与岗位本身有关，即使经过关键词提取之后，也很使得其关键词与求职保持一致，同时有些节点也会由于语义的不同而出现不同的词汇，这时再使用Jaccard相似度进行计算，就会出现语义缺失的问题，因此，接下来我们通过合成长文本输入BERT模型中来精确计算语义匹配度。

## 5.5 供求职BERT模型的建立

为了更精确捕捉求职者与岗位之间的语义关系，我们对它们的内容进行长文本拼接，并输入大型语言模型BERT模型中计算求职者与岗位之间的匹配度。这样的方法简单高效，并且能完全利用BERT模型强大的语义表达能力。

### 5.5.1 长文本的生成

长文本的生成主要是通过对文本直接进行拼接，从SQL数据库中以ID为序号提取每个ID所对应的相关属性，将每个属性前加上属性介绍，使文本更加通顺，这里以ID为“1461512488951611392”的求职者为例，他的简历如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 求职者属性 | 属性值 | 求职者属性 | 属性值 |
| 求职者ID | 1461530285551255552 | 预期城市 | 广东省,广州市,天河区 |
| 姓名 | 易女士 | 学历 | 本科 |
| 预期岗位 | 数据分析师,数据挖掘工程师 | 毕业院校 | 广西科技大学 |
| 地址 | 广东省广州市天河区 | 教育开始时间 | 2013-09-01 |
| 性别 | 女 | 教育结束时间 | 2017-06-01 |
| 求职状态 | 在职看机会 | 专业 | 应用统计学 |
| 到岗时间 | 随时到岗 | 工作企业 | 智能科技有限公司 |
| 生日 | 1993-08-22 | 工作行业 | 互联网,在线教育 |
| 工作经验 | 4年工作经验 | 工作时间 | 2017-07-01 |
| 政治面貌 | 中共党员 | 工作岗位 | 产品经理 |
| 自我评价 | 热爱数学 | 工作经历简述 | 负责线上产品策划 |
| 预期行业 | 互联网 | 关键词 | 产品策划,在线教育 |
| 求职性质 | 全职 | 语言 | 普通话:熟练,英语:良好 |
| 预期最低薪资 | 12000 | 证书 | 大数据分析工程师:2018-04-13 |
| 预期最高薪资 | 15000 | 技能 | R语言:一般,Python:一般 |

我们对该简历生成长文本如下：

易女士，意向岗位：数据分析师,数据挖掘工程师，来自广东省,广州市,天河区，性别女，可随时到岗，4年工作经验，中共党员，热爱数学，期待行业：互联网，找一份全职工作，意向工作城市广东省,广州市,天河区，本科,就读于广西科技大学,专业为应用统计学，工作公司：智能科技有限公司，所属行业：互联网,在线教育，职位：产品经理，工作内容：负责线上产品策划，岗位：产品经理，普通话熟练；英语良好，获得证书大数据分析工程师，R语言一般,Python一般。

以岗位ID为“1374177417123467264”为例，岗位信息如下：

|  |  |
| --- | --- |
| 招聘信息ID | 1374177417123467264 |
| 企业名称 | 广东泰迪智能科技股份有限公司 |
| 招聘岗位 | 新媒体运营助理 |
| 岗位性质 | 全职 |
| 薪资 | 5000-9000 |
| 支付方式 | 月薪 |
| 工作经验要求 | 1-3年 |
| 学历要求 | 本科 |
| 招聘人数 | 1 |
| 岗位福利 | 员工旅游,餐饮补贴,年终奖金 |
| 岗位职能 | 其他 |
| 岗位要求 | 岗位职责：1、负责品牌自媒体账号日常运营工作；2、负责品牌自媒体账号日常推广工作；3、负责社群（工作室）日常运营；4、完成上级交代的其他任务；岗位要求：1、工作积极，细心，有耐心；2、微信、知乎、B站深度用户，并对其传播机制有一定的了解；3、熟悉PS、PR等工具的基本操作；4、逻辑思维清晰，执行力强；5、有数据分析相关专业背景的先考虑； |
| 岗位关键词 | 在线教育,视频号运营,公众号运营 |
| 技能要求 | 产品运营 |
| 企业简介 | 教育部授权1+X大数据应用开发（Python）职业技能等级认证，工信部教考中心授权的工业和信息化人才培养工程培训基地，入选工业和信息化重点领域产业人才培训项目。 经营范围包括软件开发；软件销售；技术服务、技术开发、技术咨询、技术交流、技术转让、技术推广；软件外包服务；大数据服务；互联网数据服务；数据处理服务；数据处理和存储支持服务；信息系统运行维护服务；信息系统集成服务；信息技术咨询服务；网络技术服务；信息咨询服务（不含许可类信息咨询服务）；物联网技术研发；物联网技术服务；区块链技术相关软件和服务；人工智能应用软件开发；人工智能基础软件开发等。 |

该岗位生成长文本如下：

广东泰迪智能科技股份有限公司，简称泰迪科技，所属行业互联网,计算机软件，公司性质民营公司，企业简介教育部授权1+X大数据应用开发（Python）职业技能等级认证，工信部教考中心授权的工业和信息化人才培养工程培训基地，入选工业和信息化重点领域产业人才培训项目。经营范围包括软件开发；软件销售；技术服务、技术开发、技术咨询、技术交流、技术转让、技术推广；软件外包服务；大数据服务；互联网数据服务；数据处理服务；数据处理和存储支持服务；信息系统运行维护服务；信息系统集成服务；信息技术咨询服务；网络技术服务；信息咨询服务（不含许可类信息咨询服务）；物联网技术研发；物联网技术服务；区块链技术相关软件和服务；人工智能应用软件开发；人工智能基础软件开发等。岗位新媒体运营助理，全职，支付方式月薪，工作经验1-3年，学历要求本科以上，福利员工旅游,餐饮补贴,年终奖金，职能其他，岗位职责：1、负责品牌自媒体账号日常运营工作；2、负责品牌自媒体账号日常推广工作；3、负责社群（工作室）日常运营；4、完成上级交代的其他任务；岗位要求：1、工作积极，细心，有耐心；2、微信、知乎、B站深度用户，并对其传播机制有一定的了解；3、熟悉PS、PR等工具的基本操作；4、逻辑思维清晰，执行力强；5、有数据分析相关专业背景的优先考虑；

### 5.5.2 BERT编码与相似度计算

我们将长文本输入BERT模型中，即可对该文本进行编码。这里我们选择bert模型的sbert-base-chinese-nli模型，该模型是在预训练模型chinese\_roberta\_L-12\_H-768上微调了5个序列长度为128的epoch，通过腾讯云的UER-py工具包进行预训练与微调操作，训练数据集为ChineseTextualInference。通过输入两个句子，即可对这两个句子进行编码。并生成两个768维的向量。

通过计算这两个向量的余弦相似度，即可得到两个句子的相似度，即匹配度，这里以上述的简历长文本与岗位长文本为例。

|  |  |
| --- | --- |
| 文本类型 | 编码向量 |
| 简历长文本 | [0.8863,0.3057,...,0.3391,0.9090] |
| 岗位长文本 | [0.7819,-2.9874,...,-0.4081,0.3820] |
|  | 余弦相似度：0.6851 |

## 5.6 实验结果

通过规则匹配度模型、知识图谱模型与BERT模型，我们得到了岗位匹配度和求职者满意度的数据集，其基本结构如下：

## 5.7 小结

本章基于知识图谱与BERT模型对岗位匹配度和求职者满意度进行计算，并通过文献查找，设置自定义规则来清除匹配度为0的求职者和满意度为0的岗位，分别计算规则匹配度和规则满意度，从而根据综合评级法，对各类匹配度和满意度进行加权处理，最终得到按岗位进行分组的岗位匹配度与按求职者进行分组的求职者满意度。