文本复制检测报告单(全文标明引文)

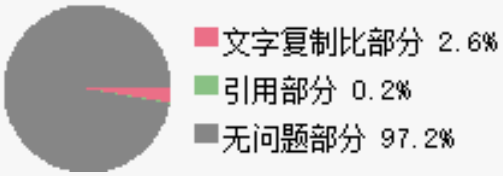
№:ADBD2018R_2018052523010543928976581 9 检测时间:2018-05-25 23:01:05

检测文献: 简历智能推荐算法
作者: 金燊
检测范围: 中国学术期刊网络出版总库
中国博士学位论文全文数据库/中国优秀硕士学位论文全文数据库
中国重要会议论文全文数据库
中国重要报纸全文数据库
中国专利全文数据库
图书资源
优先出版文献库
大学生论文联合比对库
互联网资源(包含贴吧等论坛资源)
英文数据库(涵盖期刊、博硕、会议的英文数据以及德国Springer、英国Taylor&Francis 期刊数据库等)
港澳台学术文献库
互联网文档资源
CNKI大成编客-原创作品库
个人比对库
时间范围: 1900-01-01至2018-05-25
指导教师 李舟军

检测结果

总文字复制比：2.8% 跨语言检测结果：0%
去除引用文献复制比：2.6% 去除本人已发表文献复制比：2.8%
单篇最大文字复制比：1.2%

重复字数： [641] 总段落数： [3]
总字数： [22537] 疑似段落数： [2]
单篇最大重复字数： [268] 前部重合字数： [89]
疑似段落最大重合字数：[603] 后部重合字数： [552]
疑似段落最小重合字数：[38]



指 标：☐疑似剽窃观点 ☑疑似剽窃文字表述 ☐疑似自我剽窃 ☐疑似整体剽窃 ☐过度引用

表格：0 公式：0 疑似文字的图片：0 脚注与尾注：0

6.3%（603） 简历智能推荐算法.doc_第1部分（总9571字）
0%（0） 简历智能推荐算法.doc_第2部分（总9239字）
1%（38） 简历智能推荐算法.doc_第3部分（总3727字）

（注释：无问题部分 文字复制比部分 引用部分）

1. 简历智能推荐算法.doc_第1部分 总字数：9571

相似文献列表 文字复制比：6.3%(603) 疑似剽窃观点：(0)

1	基于2D神经网络的文本问答技术研究 冯文政 - 《大学生论文联合比对库》- 2017-05-24	2.8%（268） 是否引证：否
2	梯度下降（Gradient Descent）小结 - 肖俊宁的博客 - CSDN博客 -《网络（http://blog.csdn.net）》- 2017	1.4%（135） 是否引证：否
3	基于深度学习的气体识别 陈海权 - 《大学生论文联合比对库》- 2017-04-23	0.9%（90） 是否引证：否
4	20131060047_温付洲_大规模汽车图像精细分类	0.6%（55）

温付洲 - 《大学生论文联合比对库》- 2017-05-03	是否引证：否
5 基于ASP的大学生就业招聘网站的设计与实现	0.6%（53）
刘丹;于琨;杜静翌; - 《河南机电高等专科学校学报》- 2009-11-15	是否引证：是
6 盛欣-10086226-信息管理与信息系统	0.6%（53）
盛欣 - 《大学生论文联合比对库》- 2014-06-02	是否引证：否
7 晋鑫-1310750013-基于Android的大学生就业信息发布系统的设计与实现	0.6%（53）
晋鑫 - 《大学生论文联合比对库》- 2017-05-17	是否引证：否
8 11454197_佚名_基于Android的大学生就业信息发布系统的设计与实现	0.6%（53）
佚名 - 《大学生论文联合比对库》- 2017-05-08	是否引证：否
9 基于Android的大学生就业信息发布系统的设计与实现	0.6%（53）
佚名 - 《大学生论文联合比对库》- 2017-05-16	是否引证：否
10 基于ASP的大学生就业招聘网站的设计与实现-百度文库	0.6%（53）
-《互联网文档资源（http://wenku.baidu.c）》- 2012	是否引证：否
11 基于ASP的大学生就业招聘网站的设计与实现_图文	0.6%（53）
-《互联网文档资源（http://wenku.baidu.c）》- 2016	是否引证：否
12 基于ASP的大学生就业招聘网站的设计与实现_图文	0.6%（53）
-《互联网文档资源（http://wenku.baidu.c）》- 2017	是否引证：否
13 职业技术学院人才就业网的设计与实现 - 豆丁网	0.6%（53）
-《互联网文档资源（http://www.docin.com）》- 2016	是否引证：否
14 基于BN的无人机气象威胁度建模和评估方法研究	0.3%（32）
朱国涛;周树道;叶松;王彦杰;吴家瑜; -《计算机测量与控制》- 2011-09-25	是否引证：否

原文内容　红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

单位代码10006

学号14231011

分类号

毕业设计(论文)

简历智能推荐算法

学院名称计算机学院

专业名称计算机科学与技术

学生姓名金燊

指导教师李舟军

2018年 5月 23日

本人声明

我声明，本论文及其研究工作是由本人在导师指导下独立完成的，在完成论文时所利用的一切资料均已在参考文献中列出。

作者：

签字：

时间： 2018年 5月

简历智能推荐算法

摘要

目前，越来越多不同特点的应聘者和越来越细化的岗位之间，存在巨大的信息不对称。高效、准确的将合适的人推荐到合适的岗位，有很大的实际意义。本文针对这一工程实践问题，提出“简历智能推荐算法”，将求职简历和工作表述进行匹配。

本文采用自然语言处理的多种方法，解决简历匹配的问题。首先，利用正则表达式，基于规则的提取了求职简历中，结构化字段的信息，为之后按规则筛选简历提供接口。之后，对于求职简历和工作描述的非结构化字段，提取了关键词信息：利用6000前多句带标注的数据，有监督的训练了条件随机场模型（CRF）和改进的bi-LSTM-CRF模型。分析了这两种不同模型，在技术关键词提取方面的效果。随后，分析了提取关键词的“长尾效应”，并利用自编码器无监督的学习了文本的关键词向量的表示。通过主成分分析，可视化关键词向量，结果表明匹配相同工作描述的文本的关键词向量，有聚类的现象。这一结果表明：通过自编码器确实学到了有含义的关键词向量。之后，利用多层感知机模型，对提取到的求职简历和工作描述进行分类。利用51job上提供的172封匹配的数据，训练了模型。之后，分析了模型训练的结果：在测试数据集上，画出了模型预测结果的ROC曲线，并计算了AUC。模型最终的AUC达到0.95，实现了比较好的分类效果。

关键词：自然语言处理，长尾效应，简历匹配，机器学习

CV automatically matching algorithm

Abstract

Information asymmetry is a popular problem in Talent Resource Market. Therefore, matching CV (Curriculum Vita) to its fitted job description is a important task. In this paper, I develop an algorithm to do this task. This will largely reduce the pains-taking task for people to look through CV and choose the fitted one.

The Algorithm is largely based on classical methods in Natural Language Procession (NLP). I first using Regular Expression to extract information from the structural text. This information will be used to filter the CV in the future. Then, I focus on extracting keywords from nonstructural test. Based on supervised Machine Learning algorithms, I trained an keywords extraction model. I compared two models here: Conditional Random Field (CRF) and bi-LSTM-CRF in this task. I further analyzed the keywords extracted and discovered the “long tail phenomenon”. In response to this challenge, I build an auto-encoder model to compact and extract further information from the keyword vectors of CV. Then I found that the encoded keyword vectors have rich semantics meaning: by visualizing it using Principal Component Analysis (PCA) and finding the CV’s keyword vectors clustering according to the matched job description. Finally, I build a Muti-Layer Perception (MLP) model to classify and matching CV to corresponding job description. I further analyzed the model’s performance on test set by drawing ROC curve and calculated AUC. The model achieves 0.95 in AUC, which indicated it’s capability to matching CV to job descriptions.

Key words: NLP, long tail, CV matching, machine learning

目录

“中国知网”大学生论文检测系统

1 绪论

1.1 课题背景及目的

1.1.1 现实意义

目前，越来越多不同特点的应聘者和越来越细化的岗位之间，信息不对称广泛存在。如何能够高效、准确的将合适的人推荐到合适的岗位，成为各大公司关注的重点[1]。特别是本项目关注的技术岗位的智能推荐，由于技术岗位的分化日趋复杂，人们的专业背景比较之前更加多样，各个新兴公司对于员工的要求更加多元，催生了一大批招聘网站和猎聘公司。[5] 这些网站的推送和公司的筛选，花费了大量的人力资源。根据统计：报告显示，截至2014年，网络招聘市场份额达到33.6亿元[16]如果能够设计一套更加智能的简历自动推送算法，将能够极大提高人们的工作效率，产生积极的社会影响。

1.1.2 理论意义

简历的智能推送算法，涉及到自然语言处理中的许多重要课题。简历和岗位描述，都是文本。正确的进行推送，需要挖掘文本背后的语义。因此，项目实现过程中，可能会涉及到：名称实体识别（下文缩写为NER），推荐系统（下文缩写为RS）等自然语言处理的重要课题。这里会遇到语义复杂性的问题，因此，会涉及歧义消解，属性抽取等基础理论的研究与应用来解决这些困难。

特别地，简历的文本信息中存在大量实体歧义的问题需要解决。同一个技术，同一个公司和岗位，可能在简历中出现不同的描述方式。这些歧义的消解，对于后面的推荐效果影响很大：如果实体个数多，并且存在大量未合并的冗余，就会导致后面推荐的部分，存在更加严重的“长尾”现象。这一问题，对于推荐系统的有效性，构成巨大挑战。为了应对这一挑战，本课题将会用到知识图谱的相关理论，解决信息提取时候样本稀疏的问题。

最后，该算法可以为许多类似的问题提供解决的思路。比如法律卷宗的智能检索等。

1.2 国内外研究状况

目前，有不少关于简历智能推荐的研究和应用 [2][3][4]。其中，比较普遍的模式为：搭建一个投放、展示简历的平台。有的课题在其上设计了简单的推荐算法。刘、于、杜，基于ASP实现了一个毕业生就业招聘网站。该网站，***减少了用人单位的招聘和毕业生的应聘之间不必要的限制,使得两方有了更广泛的交流,提高了双向选择的效率和成功率[4]。***罗仕鉴、陈杭渝，同样设计了同一个基于网站的平台，展示学生简历信息[3]。然而这些工作只提供了一个更快、更便捷的平台。仍然需要资深的工作人员筛选简历。在应聘者越来越多，应聘岗位越来越复杂的当下，这将耗费大量的人力资源。陈晓、王建民，提出并实现了一种新的算法，这种算法基于用户需求，对过往简历信息进行学习，建立该职位对于简历的需求模型,达到自动向用户个性化推荐简历的目。[2]然而，对于本项目针对的技术岗位招聘，由于岗位需求的多样性和简历特点的多元化，以及标注数据的匮乏，这种启发式的算法不适用。 该系统，采用了概率分类模型，分别从简历和工作描述中提取了关键实体，并对他们进行匹配。同样，对于本课题针对的复杂岗位招聘，该方法将遇到“长尾效应”等困难。

1.3 课题研究方法

本课题针对的问题是，技术岗位简历的智能推送。利用猎聘公司员工工作过程中留下的标记以及相应简历和工作描述的数据，设计简历智能推送算法。这一推荐过程，等价于预测工作描述对于简历的偏好。在预测的时候，需要进行信息的筛选和过滤，提取从文本中提取有用的信息。这与推荐系统的定义一致。因此，本课题参考关于推荐系统的设计方法。

由于简历和工作描述主要以非结构化的文本构成，因此，主要的工作重点分为两部分：文本特征的提取和特征的相似度比较。

对于本课题，文本信息特征的提取十分困难。由于技术岗位和相关技术数量庞大、种类繁多，因此直接提取的特征稀疏，难以匹配。为了解决这一困难，本文首先利用带标注的数据，训练关键词提取模型，构建关键词词表。之后采用自编码器，进一步提取和压缩，得到富含语义的关键词特征。

得到提取的特征后，进行简历推荐的过程可以参考推荐系统的若干算法（Recommendation System）。目前，推荐系统主要分成两种类型[10][11]：Content-based System (下面缩写为：CBS)和Collaborative filtering systems（下面缩写为：CFS）。由于每个工作描述对应的人和总人数相比差距很大，CFS 对应的矩阵过于稀疏，不容易训练。并且，通常新的工作描述没有任何简历与之对应，因此，CRS方法不适用。因此，基本的思路是采用CBS的方式。具体采用的方法是，利用匹配好的求职简历和工作描述的数据，经过上面介绍的提取关键词特征之后，训练一个多层感知机，完成分类任务：判断一个求职简历和一份工作描述匹配的概率。

1.4 论文构成及研究内容

本文分为五个部分：

第一部分为绪论。绪论部分介绍了问题的背景和现实意义，介绍了国内外相关领域的研究进展。并简单介绍了本文的研究思路和文章组织

第二部分为背景知识。这部分介绍了文章中使用的主要模型的原理已经相关研究。包括用来提取技术关键词的模型：条件随机场模型，和改进过的bi-LSTM-CRF模型。以及用来做无监督的特征提取的自编码器模型。最后，介绍了用来分类的多层感知机模型。

第三部分为算法的设计与实践。这部分详细介绍了建立智能推荐算法的顶层设计。并进一步介绍了数据的格式以及对数据的处理。最后，介绍了每一部分模型的结果、参数等实现细节

第四部分为算法评估。这一部分介绍了各个部分的评价标准与结果分析。依次对文章中使用的模型和方法经行系统的评估和比较。

第五部分为总结。这部分总结了全文的内容，点明文章的创新点和亮点，并展望了未来，为下一步的工作提供进一步的指导。

本文所有代码均已经上传github，供大家参考使用和指正。

（https://github.com/auas/granduate-project_auas）

2 背景知识

2.1 命名实体识别：

命名实体识别是一项具有挑战性的任务，对于输入的给定文本，输出所定义的实体的标签。比如，输入一段新闻文字，输出文中的地名实体、人名实体等。

形式上，命名实体表现为文本上的标注。比如可以给需要识别的实体，赋予特定的标签：人名标签为RN，地名标签为PN等。因此，问题转化为一个分类问题：即讨论在特定语境下，某一个单词对应若干标签的分类问题。语料来源不同，需要提取的实体不同，提取的难度也各不相同。对于结构化或者比较正式的语言，比如新闻语言，任务的难度小于一般的自然语言。

在本课题中，需要提取职业技术的关键词，因此对应的实体可以描述为一个零一的标签：0表示不是职业技术关键词，1表示是职业技术关键词。

这个任务中，最困难的地方在于，求职简历和工作描述当中的语言，更偏向日常的自然语言。存在很多元的表达，没有统一的语法和规则。

作为自然语言处理（NLP）领域的一项重要任务，命名实体识别方向有大量的相关研究。传统上需要以特征工程和词典的形式获得大量知识才能实现高性能。普遍使用的传统方法，比如应用CRF，SVM或感知器模型，依赖手工提取的特征（Ratinov和Roth，2009[19][20]; Passos等[21]，2015实现了对于自然语言中命名实体的识别任务。 然而，Collobert等人[12]提出了一个有效的神经网络，这个网络模型只需要很少的特征工程，而是从中学习重要的功能字嵌入训练大量未标记的文本 。因为，最近十分成功的，在无监督的词嵌入学习中取得进展，大量的数据（Collobert和Weston，2008; Mikolov等，2013）。

2.2 利用条件随机场的命名实体识别

条件随机场（CRF）是一类特殊的概率图模型，常用于模式识别和机器学习，并用于预测序列。条件随机场最初由Lafferty[19]提出。论文中，他指出：条件随机场与常用的隐马尔可夫模型相比，存在很多优点---包括模型不依赖强独立假设，因为从模型对于隐状态条件独立。另外，条件随机场还有效避免了，基于有向图模型的经常收到的状态出现不均的问题。状态出现不均这个问题，在很多的任务中都有出现，限制了大量模型基于有向图模型的应用。比如：在自然语言的预测标签的问题中，由于很多时候，标签存在长尾效应。长尾效应由克里斯最初在2004年首次提出，指的是尽管有些独自占比重很小，个性化很强的成分，但是他们加起来，却在整体中，占有很大的比重。比如，在网易音乐的平台上，很多人有非常个性化的偏好，这导致了许多歌曲，只有非常少的人喜欢，但是这些歌曲加起来，却吸引了超越那些非常流行的歌曲的听众。在这个命名实体识别的任务中，大量的实体在很少的句子中出现。但是这些实体，却有很大的价值：他们占了所有实体的很大的比重。传统的基于有向图的模型，比如隐马尔科夫模型，他依赖对于隐状态转移概率的建模。如果对应的标签出现的频率很低，会导致出现转移概率“假零”的问题。然而，条件随机场，是对所有隐状态的条件下，观测的条件概率建模，可以非常有效的避免这个问题。

CRF是一类基于无向图的判别模型：概率图中，有两类节点：观测节点和隐含节点。观测节点描述的是可观测的信息，隐含节点描述的是隐含的状态。在具体应用中的例子如下：比如在命名实体识别中，观测为输入的自然语言，隐含的状态是语言对应的标签，是或者不是某一个实体。再比如语音处理的任务中，观测对应输入的声音信号，隐状态对应声音信号背后的语言符号[18]。条件随机场利用概率图对于观测值和隐藏状态建模。条件随机场的具体数学定义为：

G代表条件随机长模型对于的无向图的结构。他可以通过节点“V”和节点的连边“E”刻画。每个节点表示一个随机变量，边表示两个随机变量的依赖关系。

节点由两部分组成：隐含的状态节点Y和观测状态节点X。其中X的概率，依赖Y的条件。因此模型得名：条件随机场模型。并且，条件随机场对于隐含状态Y做出来进一步的假设：满足马尔科夫性质

优化的目标函数为，观测的对数最大似然函数：

常有的优化方法是随机梯度下降的方法。这种方法，可以方便的基于图的涉及，利用已有的框架，比如TensorFlow，Keras等自动实现梯度的计算和算法的实现。本文后面采取的优化方法，就是这种基于随机梯度下降的方法。

2.3 基于bi-LSTM-CRF的命名实体识别

然而，上面介绍的条件随机场模型，输入的特征是固定的。尽管条件随机场模型本身，允许灵活的定义。但是，这些特征都是预先定义的，“静态”的特征。但是自然语言有很大的灵活性：很多词语的含义依赖复杂的上下文，这种静态的特征，并不能很好的表述词语在语句中的含义。另外，由于概率图模型本身的限制，条件随机场不能很好的刻画和捕捉这种“上下文”复杂的信息，导致对于一般自然语言的命名实体识别问题很难处理。针对这个问题，[19]提出了一种新型的神经网络架构，它使用混合双向LSTM和CNN架构自动检测字和字符级特征，从而消除了对大多数特征工程的需求。 文章还提出了一种在神经网络中编码部分词典匹配的新方法，并将其与现有方法进行比较。 广泛的评估表明，只有标记文本和公开可用的词嵌入，他们的系统在CoNLL-2003数据集上具有竞争力，并且超过了以前报道的OntoNotes 5.0数据集上的先进性能状态2.13 F1指数。文

章采用如下的模型架构：

下面将介绍模型中，动态提取文本特征的重要结构：bi-LSTM

首先，从传统的LSTM模型开始：LSTM是一种使用广泛的刻画序列的模型。传统的模型，不能刻划序列前后的依赖关系，因此不能捕捉序列内在的联系。

LSTM通过反复更新隐含层的状态，记录序列的历史信息，从而实现对于序列的建模。其核心的计算与RNN相同，公式如下:

其中ht 表示当前时刻的隐含层输出，xt 为当前时刻隐含层输入，ht-1为上一时刻隐含层的输出，W，U为权重，用于对当前时刻的信息计算。

对应如下的模型结构：

从上图可以看出，传统的RNN对应一个很深的网络，因此存在深的神经网络遇到的爆炸的梯度和消失的梯度。因此，LSTM模型对加入了门控结构，对模型引入里遗忘机制，有效的避免了这一问题。下图为RNN的示意图。

模型采用下面的更新方式：

通过使用门限控制单元，LSTM神经网络模型，让反向传播时梯度，避免了传统RNN的累乘，因此，成功避免了梯度消失问题。另外，LSTM表达力强，并且容易训练，因而在实际中得到了广泛应用。

但是，在这个任务中，关键词的判断，不仅依赖“上文的信息”，还依赖“下文的信息”。这里采用了改进过的LSTM；引入了对于下文信息的处理。这样。由于LSTM可以在实际训练过程中，动态的调节对于输入的“编码”，产生跟上下文相关的，有丰富语义的表示能够有效地提取文字的特征。因此，改进后的bi-LSTM成为一种对于传统随机向量场的改进。

2.4 自编码器的原理和应用

自编码器是一种人工神经网络，用于高效编码的无监督学习。自动编码器[18]，用来处理复杂、高维数据的降维问题 。自动编码器的目的是学习一组数据的表示（编码），通常用于降维。 最近，自编码器的概念已经越来越广泛地用于学习数据生成模型。 2010年的一些最强大的人工智能涉及在深度学习网络中堆叠稀疏自动编码器[17]。

自编码器的模型架构如下图：

自动编码器学习将来自输入层的数据压缩成短代码，然后将该代码解压缩为与原始数据非常匹配的东西。这迫使自动编码器参与降维，例如通过学习如何忽略噪声。自编码器的组成是编码器和解码器：

其中编码器的作用是，将输入的数据X映射到隐含的特征F。

解码器的作用是，将得到的隐含特征，重新映射会原始输入。

因此，总体上，模型的效果为训练如下的一个目标函数。

这个目标函数的含义是，期望编码器和解码器共同的作用下，可以还原原始的输入。这表明，得到的特征F能够很好的（无损的）包含原始的输入。

在具体的实现中，编码器和decoder是参数化的。比如，具体表现为待定参数的特定网络结构。通过上述的目标函数，（也可以使用其他不同的目标函数），利用随机梯度下降训练模型参数。

训练完成后，输入X，输出中间隐含表征F作为数据的特征。

自编码器还有其他不同的实现方式，包括可以通过在输入添加噪声，训练对于噪声鲁棒的隐含表征。

2.5 多层感知机的原理

多层感知器作为一种人工神经网络，将一组输入向量映射到另一组输出向量。可以从有向图的角度，看待MLP：它由多个节点组成，这些节点之间不是全部相连，而是形成“层的概念”，列接只存在于前一层到后一层，从输入层一直到输出层。每层直接，没有连接。这样的设计，一方面简化了模型的结果，另一方面，便于梯度下降的方法反向传播的训练参数。节点存在有一个激活函数。网络的基本结构参见下图：

每个节点，成为一个神经元。这里模拟和借鉴了生物的神经元模型：每个神经元有若干输入和输出。神经元对于输入进行处理：这里简化为对输入的信号加权求和。

模拟现实神经元的操作，每个节点还设有自己独立的偏置（bias）和“激活函数”，最后得到输出，传到下游的神经元作为下一层的输入。

其中函数f被称作激活函数，一般选用的有tanh或sigmoid函数，W表示权重，b表示偏置，W和b都是模型需要学习的参数。

对于多层的感知机，整体的结构可以用下么的公式刻画：

其中模型输入为x，W（i）分别为隐含layer和输出layer的权重，而隐含层和输出层的偏置项表示为b(i)。

MLP可以通过反向传播算法的监督学习方法训练模型的参数，通过GPU的加速，高效的拟合复杂的函数。 MLP是单层的感知器的推广，克服了简单感知器不能有效学习复杂函数的困难。

2.6 梯度下降算法

梯度下降法（gradient descent）是一种常用的，对模型进行优化求解的算法。对于样本集训练集{x(i),y(i)}，不断对模型的参数利用梯度进行更新。由于本文中，几乎所有模型的训练都依赖梯度下降的方法进行更新：包括用于提取技术关键词信息的CRF，bi-LSTM-CRF模型，以及对关键词向量进行压缩编码的自编码器和最后用于分类的多层感知机模型。这里将详细介绍梯度下降算法的原理和困难，以及随机梯度下降的改进。

在多元微积分中，对函数的各个参数求偏导数，之后用向量的形式，把得到的的各个参数的偏导数写出来，就是梯度。比如对于函数U(m,n), 对m,n分别求偏导数，之后得到的梯度向量为(∂U/∂m, ∂U/∂n)T，，称为grad U(m,n)或者▽U(m,n)。对于给定一个数据点，(t0,g0)，函数具体的梯度为(∂U/∂t0, ∂U/∂g0)T.或者▽U(t0,g0)，如果是多个参数的函数形式同理。

从几何意义看待梯度，可以理解为函数变化最快的地方。比如，对于函数f(x,y),在数据点(t0,g0)处，梯度向量的方向为：(∂U/∂t0, ∂U/∂g0)T，形象上理解，这个方向就是函数fU(t,g)变化最快的地方。也可以说，沿着梯度向量的方向变化一很小的距离，函数的增加的最多。因此同理可知：沿着梯度向量相反的方向，即： -(∂U/∂t0, ∂U/∂g0)T方向，梯度减小的最快，这也就是意味着，在这个方向上，更加容易使得函数的取值大幅下降。

下面给出梯度下降的数学形式：

函数F关于参数a可微，因此，函数F队医参数a，在其梯度方向函数值的增加最快。为了得到函数F的最小值，通过上面的式子更细参数a。其中，为一个常熟，称为“步长”。其含义为，每次更新的时候，参数a在梯度方向上，移动一个固定步长的大小。

指　标

疑似剽窃文字表述

1. 其中ht 表示当前时刻的隐含层输出，xt 为当前时刻隐含层输入，ht-1为上一时刻隐含层的输出，W，U为权重，用于对当前时刻的信息计算。
2. 梯度，避免了传统RNN的累乘，因此，成功避免了梯度消失问题。另外，LSTM表达力强，并且容易训练，因而在实际中得到了广泛应用。
3. 其中函数f被称作激活函数，一般选用的有tanh或sigmoid函数，W表示权重，b表示偏置，W和b都是模型需要学习的参数。
4. 梯度下降的改进。

   在多元微积分中，对函数的各个参数求偏导数，之后用向量的形式，把得到的的各个参数的偏导数写出来，就是梯度。

2. 简历智能推荐算法.doc_第2部分

总字数：9239

相似文献列表　文字复制比：0%(0)　疑似剽窃观点：(0)

原文内容　红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

对于一般的梯度下降算法，每次更新一个步长的时候，需要计算所有样本点。具体公式如下：

在很多实际的问题中，样本的个数N很大，每次都如此更新的话，参数的更新速度很慢，因此不适用于大规模的数据训练。于是，人们提出了几个近似的优化，简化计算的复杂度。其中就包括批梯度下降和随机梯度下降：

参数的更新方式如下：

从上面的式子，注意到，随机梯度下降在每次更新参数的时候，只计算了一个数据样本点。这样计算量变小。但随之带来的问题是，如此计算的梯度，并不是整个目标函数的梯度。由于每次只采用一个随机的数据样本点进行更新，计算得到的梯度距离真实的梯度可能差距很远。因此，人们提出了新的改进方式，即批梯度下降。其计算公式与一般的梯度下降榜示基本相同，唯一的区别在于，N不再是所有数据样本，而实每个批次里的样本。可以看出，批梯度下降是梯度下降算法和GD算法的折中。如果一个批次里面，只有一个样本，那么批梯度下降算法将退化为随机梯度下降算法。防止，如果一个批次中包含所有的样本数据点，那么批梯度下将算法将转化为一般的梯度下降算法。因此，每个批次的样本点个数，是模型训练的重要超参数之一。

从收敛性的角度看待这三种梯度下降算法，对于复杂、非凸的函数优化问题（比如多层感知机和一般的神经网络模型，都属于这种复杂的非凸函数的优化问题。一般的梯度下降算法最容易陷入局部最优，而随机梯度下降算法收敛曲线最不稳定。而正是由于随机梯度下将算法的这种不稳定，导致了它可以在一定的情况下，有效的避免局部最优和鞍点。批梯度下降算法为两者的一种折中。

3 算法的设计与实现

这个部分介绍智能推荐算法的顶层设计。主要分两部分：信息特征的抽取和分类模型的设计。对于第一部分，信息抽取，本文将其划归为两个子任务：结构化数据的抽取和非结构化的信息抽取。对于结构化的部分，通过对于求职简历中结构化字段，进行信息挖掘实现。包括做基本的数据清洗，以及通过正则表达式提取关键信息。对于非结果画的文本数据，本文再用层级信息提取的方式，一步步得到富含语义的向量表示：首先在少量带标注的数据上，训练关键词提取的模型，之后，利用这个模型，抽取无标注的求职简历中的关键词信息，得到关键词列表。之后对提取到的关键词进行简单的统计和分析，以及初步的信息筛选，完善和维护关键词列表。最后，通过自编码器无监督的学习非结构文档在关键词特征上的向量表示。并通过可视化，进行了进一步的分析。

对于第二部分，分类模型的设计，搭建了一个二分类的模型。以工作描述和求职简历的向量表示作为输入，匹配的简历结果作为标签，训练模型。前一部分实现的结果化数据，则作为以后推荐平台用来筛选使用的数据，不参与这个二分类模型的预测。

3.1 数据说明

文章中使用了多种格式的数据，分别用于模型实现和训练的各个部分。这里对于数据的使用进行统一的说明：

第一部分的数据是带技术关键词标注的文本数据。数据一共有6000句自然语言文本。每一句话经过分词得到一个词的列表。每个列表的词，对应一个{0,1}的标签，表示这个词是或者不是技术关键词。这部分数据用于训练关键词提取模型。训练好的关键词，输入一段自然语言文本，返回若干提取出的关键词。

第二个部分数据是5626分来自51job的建立数据。这部分数据用来提取关键词词表，进一步训练自编码器。这部分之后，输入一个文本，根据提取好的关键词词表生成一个技术关键词向量，文档的关键词向量在经过自编码器，得到64维度的文本的关键词向量表示。

第三部分的数据是172封匹配好的求职简历和工作描述。一共有9份不同的工作描述，每一个工作描述下面有若干匹配该工作描述的求职简历。这些配对一共有172个，这些数据作为阳性数据。之后，再通过随机组合不匹配的200个配对最为阴性数据。这些带标注的数据，用来有监督的训练多层感知机模型

3.2 数据的结构和字段分析

数据包含两部分：工作简历的数据和工作描述的数据。其中，工作描述为一段中文文本，描述岗位的相关要求：包括技能的要求以及对求职者经历的要求等。下面是一段具体的工作描述：

Web前端及UI设计

可以发现，工作描述的格式相对规则，语言也比较规范。在工作描述里面，包含了大量的关于技术的关键词。以及对于求职者工作经历的细节要求：比如要求“参与过大型java项目的开发者优先”。

工作简历为一份文档，样例参见附录一。在MongoDB中，包含的字段为：['id', 'birthday', 'skill', 'height', 'politics_status', 'crawled_at', 'location', '_id', 'education', 'gender', 'tel', 'train', 'resume_id', 'email', 'crawled_time', 'intention', 'age', 'image_url', 'work_experience', 'degree', 'household', 'award', 'marital_status', 'work', 'project', 'url', 'update_time', 'self_evaluation'}工作简历（由于不同来源的简历字段上存在细节的差异，因此，这里只展示了51job这一个来源的数据字段）。这些字段可以分成两类：一种是结构化的字段，比如性别、年龄、工作年限等。另一种是非结构化的字段，比如工作经历的描述，对参加过项目的描述。样例见下表。

结构化数据样例：

最近工作 （2年 2个月 ）

职位： 高级软件工程师

公司： ucloud云计算

行业： 互联网/电子商务

最高学历/学位

专业： 系统科学与工程

学校： 华中科技大学

学历/学位： 硕士

工作经历样例：

2015/1-至今 ucloud云计算 (2年 2个月）

互联网/电子商务|500-1000人|民营公司

关系存储开发部高级软件工程师

工作描述： 进入公司以来主导开发mysql\postgresql云数据库产品。面向开发者提供易获得、易扩容的普通、高可用数据库产品，同时带领DBA团队向用户提供产品相关的线上线下维护、咨询服务。 目前云postgresql产品是继阿里云之后，国内第二款同类产品，产品相关性能领先于阿里云. 同时负责客户需求采集及分析、竞品分析、产品设计，以及部门代码版本管理及灰度发布。 主要面向Linux服务器后台C/C++开发。涉及分布式、docker 、zookeeper、protobuffer 等开发平台。精通C/C++编程，对面向对象的编程思想、STL、设计模式等有很好的理解和掌握。熟练使用shell \python\node.js编程。 熟悉Linux系统应用编程、网络应用编程，对TCP/IP协议等有较好的了解和掌握。 熟悉Linux开发、编辑、编译、调试环境。

项目描述样例：

所属公司： ucloud云计算

项目描述： 新增云postgresql服务。 使得用户可以通过控制台或是RESTful接口快速获得托管的postgresql服务。 能够给公有云用户提供日常维护、监控、告警服务。

责任描述： 主导功能设计、开发、测试、线上发布 postgresql数据库调研，功能分析、裁剪。平衡postgresql功能的多样性和现有框架局限性之间的矛盾完成高可用方案调研、设计、开发完成分布式方案调研、设计、开发提供用户日常操作维护功能接入能力运行环境及数据库服务关键监控指标的分析、制定及实现

通过对比，可以发现：在个人的求职简历当中，特别是非结构化的字段里面，语言的用词比较工作描述而言，更加随意，属于更加日常的自然语言。在这些更加日常的文本中，存在大量歧义的词汇：比如对于技术技能的描述，存在着很多歧义和同义的表达：比如对于关键技术词汇的缩写---numpy 缩写为np。另外，还存在大量的拼写错误，比如把python 错误的贫血成

为pytohn。正是这些多元的日常语言中的表达方式，导致关键词提取时候，标签的长尾效应：即大量的标签出现频率很低，但是却非常重要，不能轻易的忽略。因为，如果粗暴的忽略了这些低频出现的标签，会导致很多关键信息的丢失。特别的，如果一个人习惯于在简历中使用与众不同的简写和缩写，那么他的求职简历当中的关键词提取就表现很差。最后的结果对于这样的求职简历是不公平的。从实际的角度看，往往是领域从业经历丰富的人，习惯于使用各种专业的简称和缩写。这样的分类系统，就容易忽略这样的潜在优秀人才。

3.3 数据的预处理

文本当中的原始数据，经过初步的数据统计后发现，存在着非常多的“非法字符”，这些字符对于后面进一步的处理不利，因此需要专门清除。由于这些“非法字符”主要是编码错误导致的乱码，可以通过python处理、过滤和清除。具体的流程如下：

①全文本切分为句子

1)利用标点符号："。|；|？|'|'|'|\\n"将短文本切分为句子

②句子切分为短句

1)利用标点：",|、|（|）|{|}|【|】|\\n|\(|\)|\[|\]"，将句子切分为短句。

③短句分词

1)利用jieba.cut将短句分词

④筛选和清理停词

1)从网上下载中文停词

2)清理停词，标点等符号

3)去除长度小于2的单词

4)去除少于两个单词的句子

除了清除“非法字符”之外，还需要为不同格式的工作简历，完成读入的接口。由于工作简历来自“51job”，“猎聘网站”，“智联招聘”等不同资源，具体的数据字段存在差异。并且，文件的格式也存在差异。其中，有.doc格式的文档，也有.html格式的数据，还有在Mongodb上面，分字段保存好的数据。因此，需要为每种格式的数据，分别设计对应的读入模块。这里在处理.doc文档的时候，由于没有方便的python接口，建立的信息都是存储在word文档的表格结构当中，并且这些网格结构都是嵌套的。（具体的情况参见附件一的样例。）针对这个困难，我利用python广度优先的便利这些嵌套的表格，利用递归的算法，实现数据的提取。

3.4 特征提取

3.4.1 基于正则表达式的格式化信息提取

对于简历中，结构化的信息，需要进一步的处理。比如在“工作年限”字段，很多人的回答是文字字符串的形式。比如“工作四年”。但是，后期做筛选的时候，需要经行数值上的比较。因此，这里利用正则表达式，匹配数字文本，进一步还原为数值形式。

这里仅仅以“工作年限”字段为例子。实际的求职简历当中，还包含着例如“期望薪金”等更加复杂的字段。比如，“期望薪金”字段，很难找到具体的规则来写正则表达式，因此，这个部分只是部分的完成。未来在这个“短文本”的信息抽取领域，还有更多的挑战有待解决。

3.4.2 技术关键词的提取

3.4.2.1 技术关键词训练数据格式说明

训练提取关键词的模型，是利用6000多条带标准的句子的训练数据来做的。训练原始数据的格式是：{x,y}其中x为一个句子，y为对应的标签。y是与x等长的零一向量，对应x中每个“单词”的标准信息。

这里对原始数据进行了进一步的处理：利用python的jieba分词工具，进一步扩展原始数据的词性标签。得到的结果为：每一个“单词”，最后具体的形式为一个二元组（单词，词性）。每一个二元组，对应一个之前的“零一标签”。这个扩充“单词语义”的部分，需要重新对原始数据进行分词。但重新分次的结果，可能跟之前的对不上。造成标签的不匹配问题。这里采用空格分隔的方式，重新拼接之前分好的单词。这样，程序会自动对这些空格的区域分开。但问题是，存在把之前分好的一个单词，拆开的情况。这里，经过统计，发现这种情况出现的非常少，因此，本文中直接将比对不上的单词删去，比对上的单词，保留之前的标签。使用这样的方法，扩充单词从字符串到元组。

3.4.2.2 条件随机场模型提取关键词

这里，采用了两种版本的条件随机场模型，进行关键词的提取。模型的实现，是依赖于python提供的一接口：pycrfsuite提供的实现方式。

第一种条件随机场模型是直接依赖pycrfsuite实现，没有自己设计模型进一步的特征表示，而是直接依靠隐含状态在图模型中的连接关系，有数据驱动的方式，直接学习相关的特征，训练关键词提取的模型。后文把这种模型的名字定义为：crf_baseline。

另外一种条件随机场模型，是自己人工进一步提取和扩充每个单词的信息。这里，我显示的引入了前后两个单词作为中间单词的特征。具体的形式如下：

一个句子可以表达为 S,w1,w2..wt-1,wt,wt+1,...wn,E

在这样的设定下，每个单词wi有前面和后面各一个单词（这里允许出现S和E）每个单词的原始格式是一个二元组（字符，词性）。这里扩城的是字符部分。将字符，扩充为一个特征列表。其中包含前后一个单词的信息。因此，调整后，数据的形式为（特征列表，词性）。特征列表的形式为[feature_f,feature_p,feature_n], feature_f, feature_p, feature_n分别对钱一个单词，当前单词以及后一个单词的特征。对于第i个单词字符串：wstr_i,提取他的特征总结在下面的表格中:

特征名称特征含义

lower 字符串的小写形式。因为存在很多英文单词，特别是技术关键词，在工作描述和求职简历当中，有多元的表达方式。其中一种最常见的，就是大小写的不同。比如python和Python，就是两种应该合并的词汇。这里提取lower这个特征，显示的处理这种特殊的“多元表达”

issuper 一个bool类型的特征，判断字符串是不是完全是大写。一般而言，一些关键词会因为大小写与一般出现的单词区分开。特别是对于关键词的提取，很多技术关键词全程很长，因此，存在将他首字母大写，再拼接起来的多元表达方式。比如，Visual Studio，在很多的简历当中，出现的形式是VS。对于这种情况，这个特征能够有效地捕捉多元的表达。

标注词性这里显示的把词性添加到单词的特征当中，强调词性特征对于模型准确预测的重要性。

3.4.2.3 bi-LSTM-CRF关键词提取

对于之前部分介绍传统CRF的的不足，这里利用文章[19]提出用来做命名实体抽取任务的模型：bi-LSTM-CRF，视图解决之前手动提取特征的一些不足。包括：特征是静态的，以及对于特征信息提取的不全面。网络架构与原来文章的相同，代码是在github上面的公开代码基础上，进行的修改和调整。

模型的整体架构如下图：

由双向连接的LSTM和CRF构成。LSTM为文本特征动态提取的工具，用来抽取文本当中的语义信息。动态的生成句子中词汇的语言信息。之所以这里采用的是双向链接的LSTM模型，是因为文本数据，不仅仅只有下文的信息对于理解当前单词有影响，上文的信息也对于理解单词含义有影响。因此，双向链接的LSTM对于文本，能有更好的提取效果。

其中，最大的改动有两个地方。一个是，之前做命名实体识别工作，该模型是一个多分类的模型，对应很多不同实体的标签。但这里做的是关键词的抽取，不区别关键词的具体小分类，因此只是一个二分类问题。

另外一个改动，是关于输入单词特征的改动。由于工作描述和求职简历的文本比较特殊，专业性比较强。另外，求职简历当中，个人化的描述存在很多非正规的表达。这导致了不能直接使用论文中已有的工具，将单词直接转化为向量表示。原始的代码支持两种不同的转换方式：一种是one-hot编码，另一种是自己添加的word embedding字典。

One-hot 编码，是先统计一共出现了多少单词，经过筛选和预处理之后，形成词典的列表。每个单词对应一个列表中的id，利用这个id完成单词到向量的转换。例如：此表中有N个单词，单词w出现在第i个位置。那么w的one-hot编码：vec_oh[w]为一个N长的向量，第i个位置是1，其他位置都是0.

word embedding的编码，是利用别人预先训练好的词向量，作为单词的数学表达。一个单词，如果没有在此表中找到对应的单词，就会根据预先的设的，赋值一个特定的向量。

经过分析发现，这两种方式都不能够直接套用。因为大量的单词在求职简历中有多元的表达，得到的词表有严重的长尾效应。对于one-hot编码，会出现大量的单词不再列表之中。而对于word embedding,大量没有比对上的单词，都会被赋予相同的数值。因此，这两种方式都不可取。

因此，这里的解决的方案是：在求职简历和工作描述中，自己构建词表，生成对应的one-hot编码，作为单词的向量表示，输入模型经行训练。

另外，值得特别说明的是，迷行采用的损失函数为：交叉熵损失函数。通过梯度下降的方法，经行训练。

模型具体的参数，参见下表：

参数名称参数赋值参数含义

batch size 64 随机梯度下降过程中，每个批的样本个数

epoch 40 最多训练多少轮

hidden_dim 300 LSTM隐状态的维度

lr 0.001 学习速率

embedding_dim 300 LSTM内部词向量word embedding的维度

3.4.3 自编码器提取one-hot编码的关键词特征

由于获得的关键词词表很大，并且里面的信息冗余，因此需要经行降维和进一步的信息抽取工作。之前步骤中，模型预测的搭配的关键词词表，维度很高，并且长尾效应严重：大量出现统一关键技术实体的不同关键词汇的多元表达。这些多元的表达可以通过简单的清洗工作，合并一些：例如，可以通过全部转换为小写字符的方式，去除Python和python这样的多元表达。

尽管如此，得到的关键词词表，依旧有很大的冗余和很高的维度。因此，我采用了自编码器来进行进一步的数据降维和处理工作。

首先，自编码器的模型架构为5个隐藏层的结构，具体的参数见下表。

参数名称参数赋值参数含义

encoder_1 500 编码器的第一层的维度,激活函数为线性函数

encoder_2 250 编码器的第二层的维度,激活函数为线性函数

encoder_3 64 编码器的第三层的维度,激活函数为线性函数

decoder_1 250 解码器的第一层的维度,激活函数为线性函数

decoder_2 500 解码器的第二层的维度,激活函数为线性函数

output 2058 输出层的维度，激活函数为sigmoid

模型的具体输入和输出是未标注简历对应关键词的向量。具体的构建方式如下：经过之前关键词模型的提取，一共在已经收集到的简历中，挖掘出K个不同的关键词，将这K个不同的关键词构成列表。第i个关键词key_i对应的向量表达是vec[key_i],其中第i个位置为1，其他位置为0。对于一个简历的文档，提取出的关键词如果为key_1....key_m（这里提取的，都是互不相同的关键词，需要过滤掉反复出现的关键词）。这些关键词对于的向量表示分别为vec[key_i]，i = 1,2...m。那么，该段文本的向量表示为：vec[doc] = vec[key_i]求和。

将vec[doc]作为输入，经过编码器和解码器之后，标准输出为原始的输入向量：vec[doc]。

自编码器的损失函数为类别交叉熵损失函数。具体的数据定义如下：

3.5 预测模型的设计

下面介绍预测模型的设计和实现细节。预测模型对于给定的两个输入的求职简历和工作描述，输入一个(0,1)之间的结果，含义为这对求职简历和工作描述比配的概率。分数越高，表明这份求职简历与工作描述的匹配程度越高。

3.5.1 模型的输入输出

预测模型的输入是工作要求（JD）和求职简历(JL)的向量。

生成向量的流程为：首先提取工作要求和求职简历当中的文本信息。进行初步的数据清洗。文本清洗包括去除无效字符，统一小写，分词，去除无用标点和符号。最后得到清晰完整的原始文本信息。

提取完文本信息后，在之前步骤得到的关键词表中，找到对应的关键词，并生成对应的文本的关键词向量。在把得到的关键词向量，输入自编码器中，得到关键词向量的编码表示。之后，将这个表示作为输入训练模型。标签为：工作介绍和求职简历是否匹配。如果匹配，则标记为1，没有匹配，则标记为0。

3.5.2 模型架构

模型的架构为多层感知机，具体的架构信息参见下面的表格：

参数名称参数赋值参数含义

hidden_1 200 多层感知机的第一层的维度,激活函数为“relu”

hidden_2 50 多层感知机的第二层的维度,激活函数为“relu”

output 1 输出层的维度，激活函数为sigmoid

3.5.3 模型训练

模型训练集大小为372，其中172个匹配的带标注阳性数据，和200个自己生成的阴性数据。阴性数据的生成，是通过把不匹配的工作描述和求职简历进行拼接得到。为了避免阳性样本点的比例过小，将生成阴性样本的个数限制在200。另外，之前自编码器输出是一个64维度的向量，对应工作描述或者求职简历的抽象向量表示。而这里的输出是工作描述和求职简历的配对，也就是这两个向量的拼接结果。得到的向量长度是128维度的。

4 推荐算法的评估

4.1 技术关键词提取结果分析

这部分，分析关键词提取的效果。首先介绍评价方式。之后，对比了三种不同模型的结果。这里一共实现了三种模型：条件随机场模型，增加人工提取特征的条件随机场模型，以及通过bi-LSTM动态提取文本特征的CRF模型。这三种模型都是命名实体识别领域很常用的模型，根据之前的介绍，技术关键词的有监督学习，可以类比一个命名实体识别的问题。因此，这里实现并对比了三种不同的模型。最后，选择了在指标上表现最好的增加人工提取特征的条件随机场模型，进行后面的关键词提取任务。

4.1.1 关键词提取评价指标

根据信息抽取任务的一般评价标准，这里引入三个指标来评价模型关键词提取的性能。他么分别是准确率（prec）,召回率（recall）和F1值。他们分别的定义参见下表：

在定义这三个评价标注之前，先引入四个符号：

符号含义

TP 真阳性个数

FP 假阳性个数

TN 真阴性个数

FN 假阴性个数

之后，依据上面定义的四个符号，进一步定义评价模型的三个指标：

符号数学表达含义

prec准确率：判断为阳性的样本中，真阳性站的比例

recall召回率：判断对的阳性样本个数，占全体阳性样本的比例

F1F1值：一个综合评价模型的指标。

3. 简历智能推荐算法.doc_第3部分 总字数：3727

相似文献列表 文字复制比：1%(38) 疑似剽窃观点：(0)

1	移动云存储安全保护方案的研究与实现 王珺(导师：李晖) -《北京邮电大学硕士论文》- 2013-12-25	0.9%（33） 是否引证：否
2	GSM网络安全协议漏洞研究 金东勋(导师：李晖) -《北京邮电大学硕士论文》- 2014-12-25	0.9%（33） 是否引证：否
3	基于Android平台移动终端安全删除的方法研究及实现 黄贤哲(导师：杨力;朱荣昌) -《西安电子科技大学硕士论文》- 2015-12-01	0.9%（33） 是否引证：否

原文内容 红色文字表示存在文字复制现象的内容; 绿色文字表示其中标明了引用的内容

综合了prec和recall,给出一个模型整体的评估。

4.1.2 技术关键词提取结果对比

指标 CRF_baseline CRF_advance biLSTM-CRF

准确率 0.737 0.772 0.712

召回率 0.746 0.850 0.859

F值 0.742 0.809 0.783

从原始的结果可以看出：改进后的条件随机场模型效果最好：综合起来的F值更高。但是，bi-LSTM-CRF没有实现预期的效果。经过具体的分析，最可能导致效果不好的原因是数据数量太小。导致最后筛选出来的模型，尽管loss比其他的模型小，但是依旧是一个比较大的数值。并且，模型最后的F值却并不理想。这提示了模型目标函数的选取不合适或者模型没有得到充分的训练。这将是下一步改进的方向。

在带标注的数据6000多句文本上面训练完模型之后，在51job的5636个求职简历数据集合中，提取了关键词。一个得到原始的关键词8251个，平均每个关键词出现23次。下图为关键词出现次数的统计图。横坐标是出现次数的对数，纵坐标是区间内关键词的个数：

从图中可以发现，有非常严重的长尾效应：将近90%+的单词，出现的次数小于5次。只出现一次的关键词，占到了整体的50%。由于提取到的关键词中，存在很多潜在的假阳性，并且很多的关键词的错误拼写被提取出来了：比如观察到的有python错误拼写成pytohon。这些都不利于后面进一步的处理。因此，这里人为的过滤掉出现频率小于3次的关键词。过滤之后，剩下了2058个关键词。下图为出现频率最高的前几个关键词的截图：从图中可以看出：出现最多的关键词是“java”,一共出现了8219次。排名前几的，大多是关于编程语言的关键词。这些技能性质的关键词，在简历中反复出现。注意到，“jsp”的出现频率也很高，达到了2063次。这个其实是Java script的缩写形式。可以看出，关键此表中，依旧存在很大量的冗余现象。存在很多关键词，虽然他们的形式不一样，但背后表示的实体其实是同一个。这些需要进一步的处理和筛选。在本文中，采取的方法是利用自编码器，来压缩特征的长度。

4.2 自编码器的实现与结果分析

根据之前的网络结果，设计并实现了自编码器。输入是原始的向量，经过编码器之后，得到了64维的文段关键词的隐含表示。

结果见下面的图：为了进一步分析和可视化，对得到的64维向量使用主成分分析，提取两个“主成分”，对向量在这主成分的投影图做可视化的散点图。下图对应的是比配过的数据的散点图。每一个类别，对应一个特定的工作表述，表明这个简历适用于这份工作描述，也就是说，他们之间匹配的程度很高。因此，同一个类别下的简历，在技术关键词层面，应该是相似的。期望在图上，会聚在一起。

实际的结果，符合这的预期：相同类别的，出现一定的“聚类的”情况。这表明模型通过关键词提取和自编码器，确实有得到更加抽象的向量表征。并且这些向量表征，能匹配上对应的实际内涵。这些结果都表明，这一系列方法对于关键词的提取是有效的。

4.3 推荐算法在测试数据集上的结果

接下来这部分介绍简历推荐算法在测试数据集的测试效果。最后的分类模型为多层感知机，下面首先介绍了模型的评价标准，之后展示和分析了模型的结果。

4.3.1 评价指标

该算法最后等价于一个二分类问题，输入求职简历和工作描述的“向量表示”（上文提到的“encoder”的输出结果），输出是一个0到1之间的实数y_pred。含义为：模型预测称该求职简历，匹配对应工作描述的“概率”。因此，这里的评价方式，是画ROC曲线和计算曲线的积分面积，即AUC值，来评估模型的分类效果。对于ROC曲线而言，如果是随机的打分，曲线为第一象限的对角线。曲线越向上，AUC的数值越大（在[0,1]区间上的积分制越大），表明模型的分类效果越好

4.3.2 结果与分析

下图为模型的ROC曲线，计算得到的AUC为0.95。这个结果表明，模型的分类效果比较理想，达到了预期的效果。（图

中的虚线为第一象限的对角线，表示的是没有任何分类效果的模型，对应的曲线）

然而，这个目前的结果仍旧存在问题：主要是带标注的数据不足，以及数据的丰富程度不够。这导致木气的结果无法直接用于实际使用。下图是带标注的数据的统计，以及每个工作描述对应的求职简历的个数。

从图中可以看出，数据集一共只有9个工作描述，一共172封求职简历。并且，这些求职简历对应的工作描述的分布非常不均匀：最多的一个工作描述，有58封求职简历，而最少的工作描述，只有6封简历。尽管在关键词部分，分析了每个工作描述对应的求职简历，提取得到的关键词向量，在经过自编码器之后，有一定“聚类”的效果，但是实际中，潜在的工作描述分类更加细致：比如同样是网页方面的工作，有不同岗位更细节的需求。

5 结论

本论文的任务是，设计简历智能推荐的算法。算法的输入是一份工作描述和一份求职简历，输出是他们匹配程度的打分。本文的工作主要是将已有的自然语言处理的方法，应用到这个问题场景，解决实际问题。主要的工作分成两大部分：特征的提取和分类与预测模型的搭建。在特征提取的任务中，对于结构化的信息，提取了关键字段的信息，进一步利用正则表达式，基于规则的提取和处理了结构化数据中的文本信息。在对于非结构化的文本信息的处理中，重点挖掘了建立文本正的关键词信息。首先，在少量带标注的数据上训练关键词抽取的模型。这一步，本文尝试了三种不同的模型，包括条件随机场的baseline模型和人工提取特征后的两种，再加上Bi-LSTM-CRF模型。在三个指标：准确率、召回率和F值的综合考量和对比下，挑选了人工抽取特征的条件随机场模型。在获得了训练好的关键词提取模型后，利用模型提取了“51job”简历数据库中，5000多封求职简历的关键信息。对提取的关键词进行了统计上的分析和初步的清洗。分析发现，抽取得到的关键词存在“长尾效应”：即大量出现频率很低的词汇，占了很大的比重。这主要是由于同一个技术实体，对应非常多元化的表达方式：比如缩写和错误拼写等问题。在数据处理中，观察到Java,jsp,java script这样的多元表达，以及类似于python和pytohon这样的错误拼写。这些问题，并没有能够通过条件随机场模型中，手动设计的参数完全解决。因此，利用自己设计的自编码器，无监督的提取了，以文档为基础单位的，关键词的抽象向量表达。通过主成分分析进行降维之后，可视化了分类后的模型。结果表明，自编码器确实学到了有意义的向量表征。表现在结果是，体现的是匹配到同一类工作描述的求职简历，最后得到的关键词向量表示聚在一起。

在完成特征抽取任务之后，进一步实现了一个二分类模型，输入为给定的工作描述和求职简历，输出为他们相似程度的打分。模型的设计采用多层感知机，将之前特征提取部分得到的，求职简历和工作描述的向量表示，进行拼接后作为输入。利用172个匹配好的求职简历和工作描述作为阳性数据，随机匹配不同类别的数据作为阴性数据，对模型进行了训练。之后，分析了结果的ROC曲线并计算了AUC。模型最后的AUC达到了0.95，表明模型有比较好的分类效果，能够比较好的预测求职简历和工作描述是否匹配。

以上是本文实现的简历智能推荐算法。由于简历库的搭建进度比预料的慢很多，最后没有来得及获取足够多带标注的求职简历的数据。尽管如此，本文实现了一个完整的流程：从数据的预处理，到特征的抽取，再到最后分类模型的训练。这些都为后面进一步的工作打下了基础：这一套完整的流程，可以启发后面对于简历匹配问题的进一步研究，并且提供了一个baseline作为参考。相信在获得更多的数据，以及对于算法进行进一步的分析之后，可以进一步提高算法的效果。

最后，值得一提的是，本文在分析和研究关键词提取的时候，发现在技术关键词抽取的领域，也出现了“长尾效应”。本文具体研究和分析了这个现象的起因，并且通过自编码器试图解决这一问题。最后得到一个还不错的结果：编码器能够学到有语义特征的信息。这表明本文的尝试有一定的成效，可以被更多的相关任务借鉴和参考。“长尾效应”是一个普遍存在的问题：特别在信息抽取和推荐系统的领域。本文采用的方法，背后的理念是：数据驱动的获得信息的抽象的分布式表示。在大量数据下，通过无监督的学习，在低维中得到这样一个更加鲁棒、更加蕴含语义信息的向量表示。笔者相信，这是在正确的方向上走出的一步，期望后续的工作，能过进一步挖掘背后的机制，并以此为基础改进算法。

致谢

感谢李舟军老师的指导。李老师在课题的选题和开题阶段，给了很多建设性意见。

感谢栾贝迪学长的帮助。学长在算法的具体设计和实现上，有很多好的建议。并且在数据集的提供和收集上予以了我很大的帮助。不仅如此，还细致的帮我批改论文、梳理思路。

感谢我的室友们帮助我梳理算法总体思路，修改论文的若干错误。

感谢家人对我的支持。忙于毕业设计的阶段，感谢家人的理解。感谢他们成为我坚强的后盾。

参考文献

[1] James, Morley-Kirk, 龚文. 中国企业招聘现状知多少——2009中国选才调查分析报告[J]. 人力资源, 2009(15):113-115.

[2] 陈晓, 王建民. 面向网络招聘的个性化简历推荐算法研究[C]// 中国数据库学术会议. 2008.

[3] 罗仕鉴, 陈杭渝. 基于Web的高校毕业生就业招聘系统的设计与实现[J]. 计算机应用研究, 2002, 19(7):135-137.

[4] 刘丹, 于琨, 杜静翌. 基于ASP的大学生就业招聘网站的设计与实现[J]. 河南机电高等专科学校学报, 2009, 17(6):122-124.

[5] 高峰. 招聘网站运营管理模式研究[D]. 北京邮电大学, 2007.

[6] Asanov D. Algorithms and Methods in Recommender Systems[J]. Berlin Institute of Technology, 2011.

[7] Mooney R J, Bunescu R C. Learning for information extraction: From named entity recognition and disambiguation to relation extraction[J]. Dissertations & Theses - Gradworks, 2007.

[8] Ingersoll G S, Morton T S, Farris A L. Taming Text: How to Find, Organize and Manipulate It[J]. Manning, 2013.
[9] Asanov D. Algorithms and Methods in Recommender Systems[J]. Berlin Institute of Technology, 2011.
[10] Kalva T R. Skill Finder: Automated Job-Resume Matching System[J]. 2013.
[11] Mikheev A, Moens M, Grover C. Named Entity recognition without gazetteers[C]// Conference on European Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 1999:1-8.
[12] Anderson C. Long Tail, the, revised and updated edition: Why the future of business is selling less of more[J]. 2008.
[13] Ratinov L, Roth D. CoNLL '09 Design Challenges and Misconceptions in Named Entity Recognition[C]// CoNLL '09: Proceedings of the Thirteenth Conference on Computational Natural Language Learning. 2009:147--155.
[14] Passos A, Kumar V, Mccallum A. Lexicon Infused Phrase Embeddings for Named Entity Resolution[J]. Computer Science, 2014.
[15] Luo G, Huang X, Lin C Y, et al. Joint Entity Recognition and Disambiguation[C]// Conference on Empirical Methods in Natural Language Processing. 2016:879-888.
[16] Collobert R, Kavukcuoglu K, Farabet C. Torch7: A Matlab-like Environment for Machine Learning[C]// BigLearn, NIPS Workshop. 2012.
[17] Li P, Huang H. Clinical Information Extraction via Convolutional Neural Network[J]. 2016.
[18] Collobert R, Weston J. A unified architecture for natural language processing:deep neural networks with multitask learning[C]// International Conference on Machine Learning. ACM, 2008:160-167.
[19] Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors[M]// Neurocomputing: foundations of research. MIT Press, 1988:533-536.
[20] Fan Y H, Chen C E, Lou L Y, et al. Design of an Optical Mechanical System for High-Resolution Encoders[J]. Applied Mechanics & Materials, 2013, 284-287:2711-2716.
[21] Fan Y H, Chen C E, Lou L Y, et al. Experiment and design of a high-resolution optical encoder[J]. Microsystem Technologies, 2013, 19(11):1775-1779.

附录

附录A 求职简历样例

说明：1.总文字复制比：被检测论文总重合字数在总字数中所占的比例
2.去除引用文献复制比：去除系统识别为引用的文献后，计算出来的重合字数在总字数中所占的比例
3.去除本人已发表文献复制比：去除作者本人已发表文献后，计算出来的重合字数在总字数中所占的比例
4.单篇最大文字复制比：被检测文献与所有相似文献比对后，重合字数占总字数的比例最大的那一篇文献的文字复制比
5.指标是由系统根据《学术论文不端行为的界定标准》自动生成的
6.红色文字表示文字复制部分;绿色文字表示引用部分
7.本报告单仅对您所选择比对资源范围内检测结果负责
amlc@cnki.net
http://check.cnki.net/
http://e.weibo.com/u/3194559873/
"中国知网"大学生论文检测系统