课程论文 2021 年装订册

"Bio Text Mining and Knowledge Discovery"

《生物文本挖掘与知识发现（本硕贯通）》[1]

（2021 年 6 月）

Jingbo Xia

2021-06

[1]A course for graduates in HZAU

Contents

1

序

这门课程源于 2016 年起开设的研究生《生物文本挖掘与知识发现概论》，每年春季授课。2020 年对生物信息专业本科生增开。2021 年起，作为本硕贯通课程开设。课程大纲和课程相关资源请在课程网站`https://hzaubionlp.com/course-bionlp-and-kd/`获取。

在介绍这份《课程论文》的内容之前，我先介绍下这门课程的基本体系。在近几年教学中，逐渐稳定成如下一个体系：

- 1 《Preface/ 序》
- 2 《Introduction of BioNLP and this Course/ 课程基本介绍》
- 3 《First Class of Linux and Lexical Analysis/ 词汇分析 Linux 基础篇》
- 4 《R programming and Word Cloud/ 词云计算和 R 调包》
- 5 《Gene Ontology (GO Enrichment and R Implementation)/ 基因本体富集分析》
- 6 《Human Phenotype Ontology (Enrichment Theory and HPO Enrichment) / 富集分析理论和人类表型本体富集分析》
- 7 《Semantic Annotation with Plant Trait Ontology/ 对作物性状本体的注释》
- 8 《PubMed Terms NER and Shell Programming/ 针对科学文献 PubMed 数据库的实体识别和 Shell 编程》
- 9 《Advanced NLP Topic in Dependency Tree and Shortest Dependency Path/ 依存关系和依存树》
- 10 《Advanced NLP Topic in Latent Semantic Analysis, from SVD to LSA/ 潜在语义分析，从奇异值分解谈起》
- 11 《A Customized Biomedical Corpus on Mutations, AGAC/ 一个定制化的生物医药领域专属语料库，活跃基因语料库》
- 12 《Advanced NLP Topic in Sequence Labeling, from HMM to CRF/ 序列标注，从隐马尔可夫到条件随机场》

- 13 《Advanced NLP Topic in Topic Modeling, from Variational Inference to Gibbs Sampling/ 主题模型，从变分推断到 Gibbs 抽样》
- 14 《Modern NLP Topic in Word Embedding, from Count-based to Prediction-based/ 基于计数的经典词表示方法和基于预测的当代词嵌入方法》
- 15 《Modern NLP Topic in Graph Embedding and Knowledge Graph, about Their Biomedical Application/ 图嵌入和知识图谱以及在生物医药领域的应用》

对于选课同学，有如下一些学习资料可资利用：一是这门课程的全程 Course Note 可在课程页面获取，网址为`https://hzaubionlp.com/data-mining/`；二是 2021 年的《课程论文 2021 年装订册》，也就是手头这份资料。通过 Course Note，读者可以全盘温习课程讲授时的 Slides 内容；通过课程论文，则可以回顾 2021 年同学们各自掌握的一些 NLP 技能。

感谢欧阳思卓和彭钱钱两位助教同学。
同样感谢 2021 年春选课的同学们。

夏静波
武汉狮子山

2

课程论文概貌

– Jingbo Xia

2.1 课程论文及其选题安排

《生物文本挖掘与知识发现概论》课程论文及选题安排

---这是课程论文选题的共享文件，所有同学都可以查看。课程论文请使用规定的Tex模板进行撰写。可以单独撰写，也可以成组撰写（2名本科生，或者1名本科生+1名研究生）。

---课程论文交稿截止时间是5月27日。论文的pdf版请发送到xiajingbo.math@gmail.com.（同日请交平时作业电子版，作业内容详见https://hzaubionlp.com/course-bionlp-and-kd/）

序号	论文题目	要求	代码难度	选题小组1	选题小组2	选题小组3
1	《生物医药语料库词汇分析》	收集和比较GENIA、AGAC、CRAFT等5个以上的生物医药文本语料库，从词汇层面分析和比较他们的异同。 --- 建议：可广泛使用TTR，词云，主题分析等方法和手段。 WordCloud代码参考：https://github.com/bionlp-hzau/Tutorial4WordCloud-Basic	★	吴明昊(单选)/ 2021.04.02 选题 **2021.04.23**预讲	杨瀚轶 王世松/ 2021.04.02选题 **2021.05.07** 预讲候选(错过)	聂有檠(单选)/ 2021.05.07选题
2	《Covid-19科学文献知识发现》	分析PubMed数据库提供的Covid-19文本摘要，围绕基因、突变、化合物等实体，进行实体抽取，对获得的实体进行知识挖掘和展示。 --- 提示：使用PubTator获取相关实体。 https://www.ncbi.nlm.nih.gov/CBBresearch/Lu/Demo/tmTools/curl.html	★★	孙阳, 黄紫嫣 /2021.04.02选题 **2021.04.23**预讲	余思克(单选)/ 2021.04.02选题 **05.07** 预讲候选(错过)**-->5.28**讲解	吴恒（单选）/2021.04.13选题
3	《水稻基因-性状的文本挖掘》	利用提供的水稻性状本体，对PubMed数据库提供的水稻文献进行挖掘，主要关注水稻基因-水稻性状之间的共句显示。 --- 提示：水稻性状本体数据：https://github.com/bionlp-hzau/TOMapping 使用PubTator获取基因实体。 https://www.ncbi.nlm.nih.gov/CBBresearch/Lu/Demo/tmTools/curl.html	★★★	孙梓淳,祝宏涛/ 2021.04.02选题 **2021.04.23**预讲	陶芳婷,黄婷婷/ 2021.04.02选题 **2021.05.07** 预讲	赵科韦,黄奇楠/ 2021.04.02选题
4	《针对AGAC数据库的序列标注》	使用序列标注工具或者神经网络代码，完成AGAC数据库的序列标注任务。并利用该任务所获得的模型，针对新文本进行挖掘。 --- 提示：需编写脚本对AGAC Track的数据进行预处理，完成Task 1. 参考论文链接：https://www.aclweb.org/anthology/D19-5710/ 参考的Wapiti项目链接：https://github.com/bionlp-hzau/Tutorial_4_CRF 参考的BiLSTM+CTF项目链接：https://github.com/bionlp-hzau/LSTM_CRF_useAGAC 参考的BERT项目链接：https://github.com/bionlp-hzau/BERT-CRF-for-BioNLP-OST2019-AGAC-Task1	★★★★★	沈放, 胡昕昀/ 2021.04.02选题 **2021.05.07** 预讲	吴思琪, 苏馨如/ 2021.05.06选题	江毅炜(单选)/ 2021.05.06选题
5	《嵌入计算》	使用Word2Vec或者BERT模型进行嵌入计算。 --- 提示：使用PyTorch框架，使用Word2Vec或者调用BERT/Transformer的深度语义嵌入模型，进行基本的语义嵌入计算，并通过t-SNE进行展示。 参考的Word2Vec项目链接：https://github.com/bionlp-hzau/Tutorial_4_word2vec 参考的BERT项目链接：https://github.com/bionlp-hzau/BERT-CRF-for-BioNLP-OST2019-AGAC-Task1	★★★★★	邓启东(单选）/ 2021.04.09选题 **2021.05.07** 预讲	刘黎（单选）/2021/04.13选题	
6	《GO/HPO富集分析》	自由选择GO富集，或者HPO富集的课题。若选用GO富集，可选用TopGO的R包，如果选用HPO富集，可选用HPO富集的往年参考代码，也可以自编。 --- 提示：请仔细揣摩课堂讲授的富集分析理论。相关代码采用R代码较为容易实现。 参考的刘彤师兄HPO富集项目：https://github.com/tongliu-liu/HPO-enrichment-analysis 参考的GO富集项目：https://github.com/bionlp-hzau/Tutorial_4_GO_Enrichment	★★ GO富集 / ★★★★ HPO富集	杨晓龙, 周旺/ 2021.04.09选题 **2021.04.23**预讲	晏倫一（单选)/ 2021.04.09选题 **2021.05.07** 预讲候选(错过)	杨迟 屈伸洋/ 2021.04.09选题
7	《老年痴呆症(Alzheimer's disease)科学文献的核心词汇、语法和依存路径分析》	分析PubMed数据库提供的AD文本摘要，围绕基因、突变、化合物等核心词汇的语法和依存路径，结合所挖掘的生物实体，进行知识挖掘和展示。 --- 提示：使用PubTator获取相关实体。 https://www.ncbi.nlm.nih.gov/CBBresearch/Lu/Demo/tmTools/curl.html 使用依存数信息，参考项目链接：https://github.com/bionlp-hzau/tutorial4dependencytree	★★★	张富卿,李佳桐/ 2021.04.17 选题 **2021.05.07** 预讲 (错过)**-->5.28**讲解		

2.2 论文要求和评分依据等

我们在论文模板中明确给出了课程论文的要求。

- 课程论文不同于一般的学术论文，并不强调突出 Novelty 或者 Contribution，而是**强调反映学生在课堂所学所思，强调原创，鼓励观点和经验上的“主观性”**。

- 同时，课程论文也对撰写的章节安排有硬性要求，**在使用此模版进行论文撰写时，请删除说明部分，但务必请保留本模板中的所有 section 和 subsection 设置，并按照每个章节的撰写要求进行撰写**。
- **第 4 和第 7 章 (即第 4、7 个 section) 视情况需要可以整体删除，6.3, 6.4, 6.5 这些 subsection 可以依情况删除；除此之外，其余 section 及其 subsection 一律必须保留；仅允许在第 5 章自由增加 subsection；允许在任意 subsection 中适量增加 subsubsection**。
- 所选择标题必须与分配的论文标题一致，可以在后面增加小标题。课程论文依据任务要求进行。因为不同的题目在代码要求上有一定的差异，代码原创性较多的论文，可以详述其算法部分的理解，略介绍其后的生物信息实验；代码份额较少的论文，则可在后期生物信息实验和知识挖掘上多下一些功夫。
- 请务必使用中文撰写!!!
- 不鼓励粘贴过多代码
- 非常鼓励将代码和项目完整地上传到 GitHub，并在论文中提供访问链接。
- 除参考文献外，论文页码在 4-7 页之间。严格控制版面。请使用默认的页边距、行距和字号。

更详尽和完整的课程论文要求请下载模板 tex 文档 (`https://hzaubionlp.com/course-bionlp-and-kd/`)。

2.3 初版后修订要求

2021 年 5 月 27 日初版后修订要求：

- 错别字修订，例如 “的/地/得”，“HHPO”；
- 务必使用课程网站指定的 Latex 模板；
- 务必使用参考文献！
- 使用 bib 来编纂参考文献，选用本地软件如 TexWorks 请记得编译 BibTex 以生成参考文献 / 也可使用 Overleaf 网站，其提供 bib 自动编译功能；
- 图片给出出处：使用 cite{ } 命令引用来源文献，若自绘，则介绍绘制方法；
- 用足 7 页篇幅（不包括参考文献），不允许超页；
- 不允许改变字号和行距；
- 代码和数据在文中给予恰当说明，相关项目上传到 GitHub，文中给出 GitHub 项目地址；
- 注重项目的 Reproducibility 和 user-friendly；
- 修订版发送 xiajingbo.math@gmail.com，截止时间：6 月 4 日晚 11 点 59 分。

2.4 论文列表

论文列表

- 吴明昊《生物医药语料库词汇分析》
- 杨瀚轶、王世松《生物医药语料库词汇分析》
- 聂有攀《生物医药语料库词汇分析》
- 晏倫一《HPO 富集分析》
- 杨晓龙、周旺《HPO 富集分析》
- 杨迟、屈伸洋《HPO 富集分析》
- 孙梓淳、祝宏涛《水稻基因-性状的文本挖掘》
- 赵柯韦、黄奇楠《水稻基因与性状的共句显示》
- 陶芳婷、黄婷婷《水稻基因-性状的挖掘》
- 吴恒《Covid-19 科学文献知识发现》
- 余思克《Covid-19 科学文献知识发现》
- 孙阳、黄紫嫣《Covid-19 科学文献知识发现》
- 张富卿、李佳桐《老年痴呆症科学文献的核心词汇，语法和依存路径分析》
- 吴思琪、苏馨如《对 AGAC 数据库的序列标注任务与新文本挖掘》
- 江毅炜《针对 AGAC 数据库的序列标注》
- 胡昕昀、沈敖《基于 BiLSTM 和 CRF 的序列标注》
- 邓启东《Word2Vec 及 BERT 模型的水稻文献词汇嵌入计算》
- 刘 Yun《基于 Bert 的整合水稻性状本体 (RTO) 的嵌入和展示》

3

生物医药语料库的词汇学特征探索

> 在这个部分，我们请同学们用一些基本的词汇学探究方法，调查 *GENIAl*、*CRAFT*、*AGAC* 外加 2 个其他的生物医药专属语料库的词汇学特征和区别。
>
> – Jingbo Xia

项目要求：

收集和比较 GENIA、AGAC、CRAFT 等 5 个以上的生物医药文本语料库，从词汇层面分析和比较他们的异同。

建议：可广泛使用 TTR，词云，主题分析等方法和手段。

WordCloud 代码参考：`https://github.com/bionlp-hzau/Tutorial4WordCloud-Basic`

相关论文三篇：

吴明昊《生物医药语料库词汇分析》

王世松杨瀚轶《生物医药语料库词汇分析》

聂有攀《生物医药语料库词汇分析》

3.1 吴明昊《生物医药语料库词汇分析

关于生物与医药的研究是一个永恒的话题。本文是基于语料库研究的方法，运用 TTR(Type Token Ratio)、词云、主题分析等手段，统计分析了当下常用的 OSIRIS、LLL、GENEREG、IEPA 以及 GETM 这 5 个语料库中的词汇丰富度以及词频，并结合当下生物医药的研究现状分析目前语料库中主要的研究方向。

生物医药语料库词汇分析

吴明昊 [1]

[1] 华中农业大学信息学院，430070，武汉，湖北，中国

摘要

关于生物与医药的研究是一个永恒的话题。本文是基于语料库研究的方法，运用 TTR(Type Token Ratio)、词云、主题分析等手段，统计分析了当下常用的 OSIRIS、LLL、GENEREG、IEPA 以及 GETM 这 5 个语料库中的词汇丰富度以及词频，并结合当下生物医药的研究现状分析目前语料库中主要的研究方向。

关键词：语料库、TTR、词云、主题分析、OSIRIS、LLL、GENEREG、IEPA、GETM

1 课题概况

生物医药行业的快速发展基于生物科技水平的不断进步，同时其也是使生物制药成为我国重要经济支柱的原因之一 [1]。但是，我国的生物科技与国际上其他发达国家的生物科技水平相比还存在着不小的差距，生物医药行业也还存在许多问题有待改善。据此有必要对当下行业现状进行分析，以此来对该行业的未来发展趋势提供经验和方向。

1.1 国内行业现状

近年来，医药产业营业收入呈上升趋势。2019 年，仅河北医药产业营业收入和利润，分别较 2018 年增长了 12.3% 和 22.1%，工业增加值增长比率占全省的比重为 2.6%。关于医药投资方面，2019 年，河北规模以上医药企业的医药研发费用较上年增长 37.0%，其占全省的研发费用比重为 8.73%[2]，高于营业收入的比重。

中国市场以做仿制药为主，原研药很少，生物药占国内医药市场大概 20%，其中单抗类药物是治疗肿瘤病症的非常有效的生物药，规模越来越大，远远高于其他的生物医药产品，是带动着中国基因工程药物发展的领头羊，但总体来说与国际水平仍有很大差距。

1.2 国外研究现状

治疗领域集中体现在：目前的产品治疗领域还十分单一，几个畅销的生物药物占据了主要市场份额。由美国市场调研机构 EvaluatePharma 最新出具的预测报告显示，2018 年最畅销与销售额增长最大的 TOP10 药物全为生物药，单抗类药物占 8 种。仅从单抗未来销售飘红，占据八成市场的形势来看，抗体药类药物已成为生物药中的王牌，也是各巨头必争之地。单抗药物应用日趋广泛，已经成为肿瘤靶向治疗的主流用药。全球单抗药物市场规模已从 2005 年的 140 亿美元增长到 2010 年的 510 亿美元，年复合增长率达 30%，在生物药中的占比超过 30%，成为生物药中占比最大的品类 [3]。

2 数据

本文采用了 OSIRIS、LLL、GENEREG、IEPA 以及 GETM 这 5 个语料库，数据来源于网站 http://corpora.informatik.berlin.de/，该网站有关于语料库具体的研究事项和研究方向说明。

但是拿到数据之后，需要对数据进行提取处理，文件中所含有的.txt 文件无法直接使用，其中包含了大量的无效字段，需要编程进行提取。

3 研究方法

本研究采用语料库语言学的方法，对在 OSIRIS、LLL、GENEREG、IEPA 以及 GETM 这 5 个语料库中的词频以及丰富度进行分析，如表 1 所示为这五种语料库的主要内容。主要运用的分析方法包括 TTR(Type Token Ratio)、词云、主题分析这三种方式。在国际生物医药研究方向的背景下，针对语料库中的高频词进行抽取分析。

表 1: **五种语料库的主要内容**

Corpus name	Main content
OSIRIS	The OSIRIS corpus is a set of MEDLINE abstracts manually annotated with human variation mentions.
LLL	Annotation indicating agent and target of a gene interaction.
GENEREG	regulation of gene expression
IEPA	protein-protein interactions
GETM	GETM is a tool which is capable of extracting information about the expression of genes from biomedical literature. Using the data extracted by GETM.

.

3.1 Type Token Ratio

TTR 在不同类型的文体中有着不一样的数据，若以其中“文学”的数据为基准，其他作者的文本的 TTR 超过这个基准数据，那意味着作者在写作的时候用词丰富性不够、重复内容较多、语言较通俗易懂等，低于这个基准数据，则意味着作者用了一些生僻的词语、用词丰富性很高等。如图 1 所示。但是由于 TTR 算法太基础并且在本次研究中作用可能不大，故不采用。

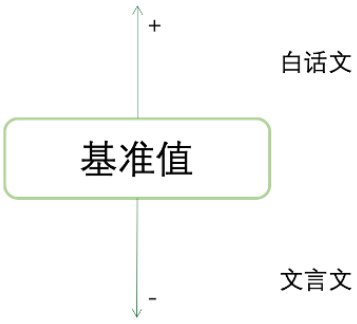


图 1: TTR 基准定义

3.2 词云

通过形成“关键词云层”或“关键词渲染”，对网络文本中出现频率较高的“关键词”的视觉上的突出。如图 2 词云示例图所示，对一本日记进行词云分析，可以看到出现最多的词语可能就是文本中的重点内容，但从词频的角度来看，还是需要进行甄别。

图 2: 词云示例图

3.3 主题模型分析

对文字中隐含主题的一种建模方法，主题就是一个概念、一个方面。它表现为一系列相关的词语，对模型进行归类。当我们提到“百度”“李彦宏”的时候就会自然联想到百度公司，提到“Windows”“操作系统”的时候就会自然联想到微软公司一样，主题模型分析是在提取文段中的隐含主题时常用的一种方法，其中 LDA 在主题模型中占有非常重要的地位，常用来文本分类。如图 3 主题模型视图分析所示。

图 3: 主题模型视图分析

LDA 中涉及到的先验经验：二项分布、GAMMA 函数、BETA 分布、多项分布、Dirichlet 分布、马尔科夫链、MCMC、GibsSampling、EM 算法等。在本文中我会根据自己的理解说明。

4 算法说明

4.1 二项分布与多项分布

二项分布是 N 重伯努利分布，即为 X B(n,p). 概率密度公式为：

$$P(K=k)=\binom{n}{k}p^k(1-p)^{n-k}$$

二项分布扩展到多维的时候就是多项分布，相对比于二项分布，单次试验的随机变量取值不再是 0-1，而是有多种离散值的可能 (1,2,3⋯,k) 概率密度函数为：

$$P(x_1,x_2,x_3,\dots.,x_k;n,p_1,p_2,p_3\dots,p_k)=\frac{n!}{x_1!\dots x_k!}{p_1}^{x_1}\dots{p_k}^{x_k}$$

4.2 Gamma 函数

Gamma 函数的定义：

$$\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt$$

分布积分后，可以发现 Gamma 函数有这样的性质：

$$\Gamma(x+1) = x\Gamma(x)$$

Gamma 函数可以看成是阶乘在实数集上的延拓，具有性质：

$$\Gamma(n) = (n-1)!$$

4.3 Beta 分布

Beta 分布的定义：对于参数 alpha > 0, beta > 0, 取值范围为 [0, 1] 的随机变量 x 的概率密度函数为：

$$f(x;\alpha,\beta) = \frac{1}{B(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1}, \quad 其中 \frac{1}{B(\alpha,\beta)} = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}$$

4.4 共轭先验分布

共轭的意思是，以 Beta 分布和二项式分布为例，数据符合二项分布的时候，参数的先验分布和后验分布都能够保持 Beta 分布的形式，这种形式的好处是，能够在先验分布中赋予参数明确的物理意义，这个物理意义能够延续到后续分布中进行解释，同时从先验变换到后验的过程中，从数据中补充的知识也能够有物理解释。

4.5 Dirichlet 分布

Dirichlet 的概率密度函数为：

$$f(x_1, x_2, \ldots, x_k; \alpha_1, \alpha_2, \ldots, \alpha_k) = \frac{1}{B(\alpha)} \prod_{i=1}^{k} x_i^{\alpha^i - 1}, \quad 其中 B(\alpha) = \frac{\prod_{i=1}^{k} \Gamma(\alpha^i)}{\Gamma(\sum_{i=1}^{k} \alpha^i)}, \sum_{i=1}^{k} x^i = 1$$

根据 Beta 分布、二项分布、Dirichlet 分布、多项式分布的公式，验证了 Beta 分布是二项式分布的共轭先验分布，而狄利克雷分布是多项式分布的共轭分布。

4.6 LDA 解析

在查阅了相关文献之后，我对 LDA 有了一定的理解，例如文档中每个词的生产都需要两个随机数，第一个 doc-topic 随机数得到 topic，第二个 topic-word 随机数的到 word，每次生产每篇文档中的一个词的时候这两个随机数的是紧邻轮换进行的。如果语料中一共有 N 个词，则一共要有 2N 个随机数，轮换的产生 doc-topic 和 topic-word 的随机数。

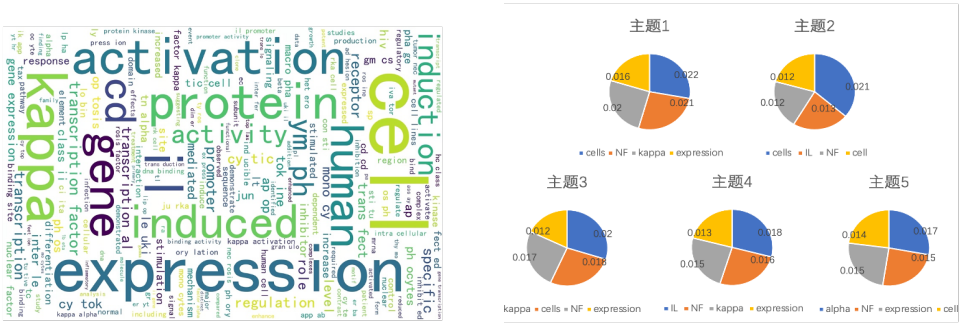
但是实际上有一些随机数的产生顺序是可以交换的，可以等价的调整 2N 次随机数的产生顺序：前 N 次只产生 doc-topic 随机数得到语料库中的所有 topic，然后基于得到的每个词的 topic 编号，后 N 次 topic-word 随机数生成 N 个 word，此时可以得到：

$$p(\vec{w},\vec{z}|\vec{\alpha},\vec{\beta}) = p(\vec{w}|\vec{z},\vec{\beta})p(\vec{z}|\vec{\alpha})$$
$$=\prod_{k=1}^{K}\frac{\Delta(\overrightarrow{\theta_K}+\vec{\beta})}{\Delta(\vec{\beta})}\prod_{m=1}^{M}\frac{\Delta(\overrightarrow{\theta_m}+\vec{\alpha})}{\vec{\alpha}}$$

5 主要的生物信息实验和实验结论

在运行代码前，需要对数据进行的预处理，将语料库中的 stoplist 进行了更新，其中包括一系列生物相关的停用词库以及常用的英文词，对词库进行筛选整合，形成了适合本实验的 stoplist，对后续实验的各项分析的精度起到了关键性的作用。

5.1 GETM 语料库

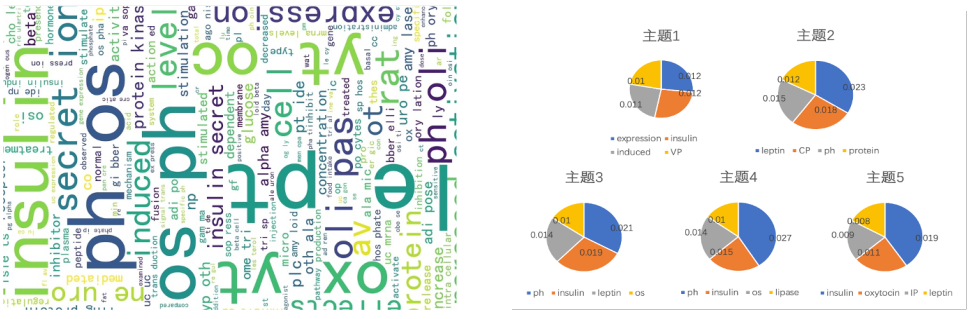


(a) GETM 语料库词云分析　　(b) GETM 语料库主题分析

图 4: GETM 语料库分析

GETM 语料库的词云分析如图 4(a) 所示，主题分析如图 4(b) 所示。根据图 4(a) 的词云分析实验结果，可以清楚的发现在 GETM 语料库中，“gene”“cell” 为主要出现的非停用词词语，且在主题分析中，“cell” 几乎出现在所有的主题中且占据四分之一左右的比例，由此可以看出，大部分基于 GETM 语料库的直接实验以及衍生实验，其相关的生物基础以及背景是与 “cell” 相关。

5.2 IEPA 语料库



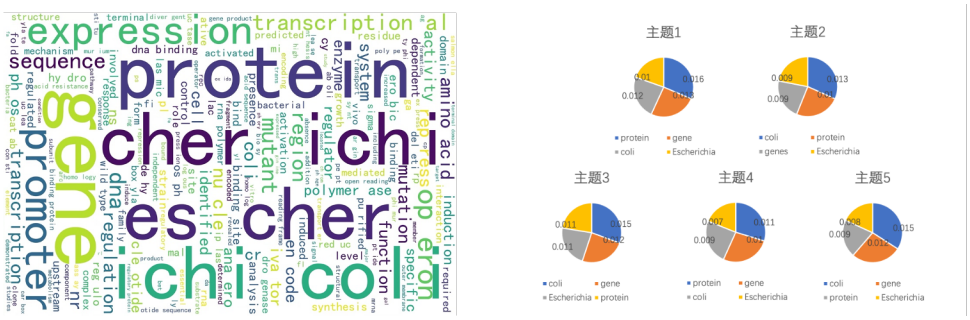
(a) IEPA 语料库词云分析　　(b) IEPA 语料库主题分析

图 5: IEPA 语料库分析

IEPA 语料库的词云分析如图 5(a) 所示，主题分析如图 5(b) 所示。从图 5(a) 中可以看出 “insulin” 胰岛素、“leptin” 瘦蛋白、“phos” 磷酸酯酶、“osyt” 催产素都属于类似催化剂一类，在蛋白质合成或者分解

过程中，催化剂的作用毋庸置疑，而且图 7 的主题分析中可以看出，无论是哪一种主题都涉及到一种或者两种激素类物质，由此可以看出 IEPA 语料库围绕的核心研究方向是蛋白类物质以及蛋白催化合成等相关领域。

5.3 GENEREG 语料库

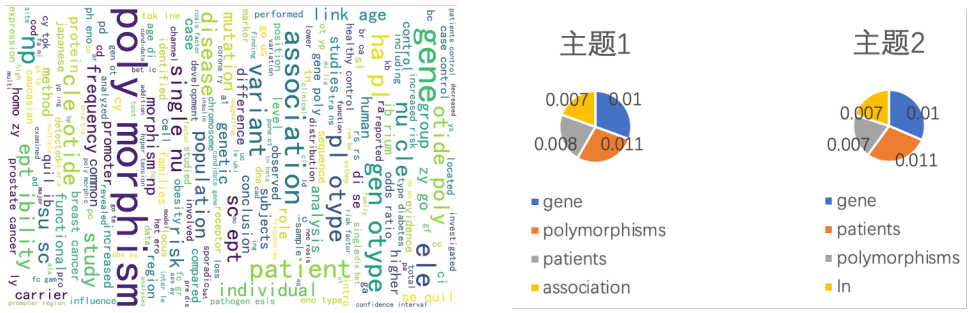


(a) GENEREG 语料库词云分析　　(b) GENEREG 语料库主题分析

图 6: GENEREG 语料库分析

GENEREG 语料库的词云分析如图 6(a) 所示，主题分析如图 6(b) 所示。从词云分析图中，可以清楚的看到 “gene” 基因、“promoter” 启动子、“Escherichia” 大肠杆菌、“protein” 蛋白质等高频非 stoplist 中的词，可以大致推断出 GENEREG 语料库是围绕 Escherichia 大肠杆菌的研究，而在 GENEREG 语料库的主题分析中，所有的主题都包含有 “Escherichia” 以及 “gene”，从而可以得出结论，该语料库的研究方向是有关于 Escherichia 大肠杆菌的基因编辑，通过对 promoter 启动子和 protein 蛋白的编辑从而改变大肠杆菌的相关属性。

5.4 OSIRIS 语料库

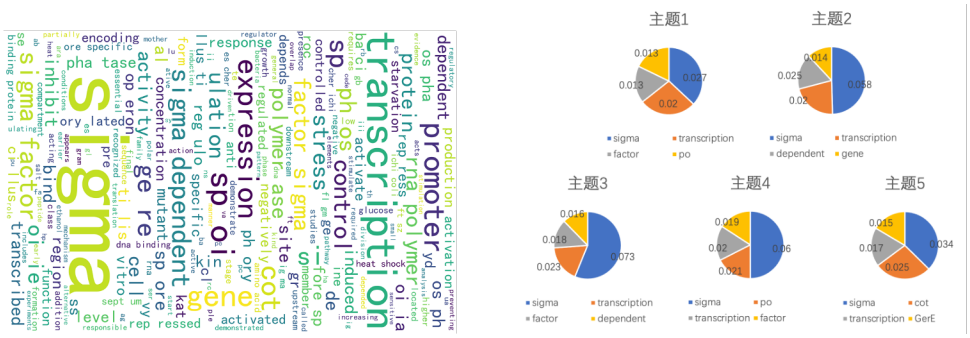


(a) OSIRIS 语料库词云分析　　(b) OSIRIS 语料库主题分析

图 7: OSIRIS 语料库分析

OSIRIS 语料库的词云分析如图 7(a) 所示，主题分析如 7(b) 所示。在 OSIRIS 语料库中的词云分析中 “genotype” 基因型、“gene” 基因、“polymorphism” 多态为非停用词的高频词，可以看出 OSIRIS 语料库中的主要内容与基因有关，从主体分析中可以得出 OSIRIS 语料库的研究对象与 “patients” 有关，涉及到的具体方向为与病人相关的基因研究，并且与基因之间的 “association” 协作表型有关。

5.5 LLL 语料库分析



(a) LLL 语料库词云分析　　(b) LLL 语料库主题分析

图 8: LLL 语料库分析

LLL 语料库的词云分析如图 8(a) 所示，主题分析如 8(b) 所示。“sigma factor” 西格玛因子也称为务马因子，其是一种专一性蛋白，作为所有 RNA 聚合酶的辅助因子起作用，从词云分析图中可以看出 LLL 语料库与 “transcription” 转录以及 “gene” 基因有着密切联系，而 LLL 语料库的主题分析中，“sigma” 以及 “transcription” 同样存在于每个主题中，进一步验证了之前的结论。

6 后记

在第 4 章中对于 OSIRIS、LLL、GENEREG、IEPA 以及 GETM 这 5 个语料库进行了分析，主要运用了词云和主题分析，从分析结果可以看出，当下生物医药的主要研究方向逐渐着眼于 gene 以及 protein 层次，并且每个语料库都趋于专一化，只针对特定的蛋白或者品类进行研究，由此可见，在很长一段时间内，生物医药会一直停留在 gene 水平，如果能够发现比 gene 更小的组成因子，才会进一步发展。而对于语料库的发展，在未来的一段时间内，都会呈现出专一性，研究会逐步缩小到某个基因，而不是单个个体的基因。

6.1 课程论文构思和撰写过程

在设计论文阶段，查阅了部分资料，关于 LDA 算法的原理比较模糊，只能勉强看懂讲解，词云和 TTR 原理算法相对简单，阅读起来比较容易。拿到课程题目的时候，个人的大致思路如下：

(1) 锁定环境背景，将课程论文放在有特定方向或者说是环境下进行构思，这样更容易了解到自己需要做什么和完成什么任务。经过考虑，决定将背景选定为生物医药当下的研究环境。

(2) 锁定了背景之后，就开始查找相关的文件综述，但是很遗憾，由于个人能力有限且研究方向为计算机视觉，对文本处理的了解不多，所以开始的查找方向仅仅限于生物医药的研究，没有跳出思维限制，作为一名计算机的学生，应该对算法有敏锐的嗅觉，从开始就应该直奔主题，查找与 TTR、词云、主题模型分析等有关的文章进行学习，这是今后学习中应该注意的地方。

(3) 在阅读文献之后，我开始构思全文的基本思路，我个人觉得课程中给出的论文模板不适合所有的选题，所以我部分章节考虑使用了自己的思路进行撰写。按照我的最初思考，文章的章节应该为：摘要、引言、国内外相关工作及选题意义、研究方法、实验结果分析、总结。再者，从思维局限性和思维延展性的方面考虑，文章中的小标题不应该规定死，这样不利于学生独立思考分章，因为小标题相当于个人的行文和思考，限定死了对有想法的学生会形成思维定式，没有想法的学生又会强行凑字。

（4）撰写过程中遇到许多困难，首先就是知识盲区，因为主题背景设定为当下生物与医药的大背景，而个人又没有相关的生物基础，拿到数据之后往往不知道如何分析，对于生物医药相关的专有名词不够敏感。其次是技术盲区，第一次拿到语料库进行实验的时候，跑出来的结果不尽如人意，后来通过请教同实验室文本处理的同学，才知道需要对数据进行提取，还要增加 stoplist 停用词表格。最后就是排版工具的使用，老师给的模板在 overleaf 在线编辑网站上没法运行，在 TeXWorks 上面也 run 不通，最后是自己仿照老师给的模板重新用 TeXWorks 编写的，因为之前一直使用 Word 文档，所以花费了大量的时间在编写排版上。建议直接使用相关顶刊的格式要求。

（5）对于课程安排，老师相当负责并且学术能力很强，并且我在这堂课上看到了许多我们以前本科的时候没有的东西，华农现在的本科生们很强，比我们那个年纪更有激情，这是最让我感动的地方。

6.2 所参考的资源来源

我参考的资源来源了几个方面，首先就是课堂，在课堂上老师讲的知识包括两次研究生分享，握都有很大的收获。其次就是实验室的同袍，在文本处理这个方向上积累了大量的知识，可以给我论文撰写的灵感和建议。再者是欧阳思卓和王师弟，在撰写过程中的细节上给予了我很大的帮助。最后就是互联网，大部分语料库在 http://corpora.informatik.hu-berlin.de/上都能找到，而且会有详细的说明，但是要挂梯子，然后关于算法理解部分，参考了知乎、CSDN 上的讲解。

7 参考文献

[1] 娄霁月. 我国生物医药行业现状与发展趋势 [J]. 大众标准化 2019(16):61-62.

[2] 米彦泽，曹思宁. 一季度我省医药行业增加值同比增 6.2% [N]. 河北日报，2019-05-17（02）.

[3] 沈庆澜, 生物药产品开发及市场分析 [J]. 现代商业, 2020(5):79-80.

3.2 王世松、杨瀚轶《生物医药语料库词汇分析》

语料库是存储在计算机上，用于研究语言是如何使用的书面或口头的自然语言材料集合。更准确地说，语料库是用于语言分析和语料分析的系统化和计算机化的真实语言集合。为了开发 NLP 应用，我们需要书面或口头的自然语言材料作为语料库。这些材料或数据被用作输入并帮助我们开发 NLP 应用。语料库是 NLP 相关应用中最关键，最基本的部分，它提供了用于构建应用的定量数据。语料分析是一种以真实上下文和交际语境为基础的，深入研究语言的方法。本节讨论的是数字化存储，可以通过计算机获取、检索和分析的语料库。

课程论文 GitHub 网址：`https://github.com/Bing-nai/ziran_code.git/`

生物医药语料库词汇分析

杨瀚轶[1], 王世松[2]

[1]华中农业大学信息学院，430070，武汉，湖北，中国
[2]华中农业大学信息学院，430070，武汉，湖北，中国

摘要

语料库是存储在计算机上，用于研究语言是如何使用的书面或口头的自然语言材料集合。更准确地说，语料库是用于语言分析和语料分析的系统化和计算机化的真实语言集合。为了开发NLP应用，我们需要书面或口头的自然语言材料作为语料库。这些材料或数据被用作输入并帮助我们开发NLP应用。语料库是NLP相关应用中最关键，最基本的部分，它提供了用于构建应用的定量数据。语料分析是一种以真实上下文和交际语境为基础的，深入研究语言的方法。本节讨论的是数字化存储，可以通过计算机获取、检索和分析的语料库。

关键词: NLP，语料库，TTR，词云

1 课题概况

文本数据的语料分析包括对数据集的统计调查、操作和泛化。对文本数据集，通常是对语料库进行出现多少不同的单词，单词频率分别是多少的分析。几乎每一个NLP应用开发都需要进行一些基础的语料分析来帮助我们更好地理解语料库。

近些年来，深度学习，人工神经网络在我们生活中频繁出现，NLP也发展到了新高度，我们粗略的了解NLP分析的一些流程，认为对语料库的处理和分析是NLP后续操作的基础，这对于我们未来NLP的扩展学习和同专业知识结合有较大的价值。我们也通过对生物医学语料库的分析，掌握了文本处理的一点基本方法。

2 数据

为了对比生物医学类语料库与生物类著作在用语上的差异，本次试验我们共收集了五个语料库，分别是三个生物医学类语料库和两个我们自己构建的生物书籍类语料库，生物医学类语料库包括AIMED、GENEREG、IEPA，生物书籍类语料库包括由《物种起源》、《GENEX》、《生命是什么》构成的语料库，我们定义为严谨型生物类书籍语料库，由《昆虫记》、《昆虫记忆》、《自私的基因》构成的语料库，定义为科普型生物类作品语料库。

3 研究方法

3.1 研究方法的算法背景，与其他方法的联系与区别

NLP是从1950年图灵测试才出现，图灵测试指的是通过人与机器交流让人判断交流的是人还是机器，验证机器是否智能。在1950-1970年间，NLP的主流是基于规则形式的语言理论，科学家根据数学中的公理化研究自然语言，试图用有限的规则描述无限的语言现象。1970年至今，NLP的主流是基于统计，Google机器翻译打败了基于规则的Sys Tran [1]。2010年后，机器学习逆袭成为主流，AlphaGo掀起了人工智能潮。

3.2 研究方法中的核心思路

我们搜索资料，了解到NLP的一般流程主要包括获取预料，语料预处理，特征工程，特征选择，模型训练，评价指标以及模型上线应用，这次实验我们主要进行的是前两个步骤。通过语料清洗，分词分句处理，去停用词等计算语料库的TTR，绘制语料库的词云以及对语料库的句子进行比较 [2]。

3.3 本文的方法部分与课堂讲授内容的联系和区别/或补充

本文的基本代码与课堂讲授内容几乎没有差别，但是在运行过程中为达到不同目的有少量修改，对于整体实验的思路，我们也是在课堂讲授的基础上进行拓展并自己加以分析与推测。我们使用到的公式是TTR=$\frac{amount\quad of\quad unique\quad words}{amount\quad of\quad total\quad words}$，用以计算词汇的多样性 [3]。

4 算法实践和代码编写要求

4.1 任务描述

收集5个生物医学类语料库，对这5个语料库进行一系列分析，分析包括课堂所讲述的TTR与词云，以及尽可能拓展分析。

4.2 实验设计

此次实验使用了Ubuntu 20.04，R-4.0.5，Python 3.7，在Linux中主要进行的是对数据的预处理，在R中主要进行的是检验与绘图，在Python中仅使用了自己编写计算句子词数的代码。对于实验数据的预处理，我们进行了分词处理，词形还原和分句处理。

在Linux中，通过将语料库的非字符全部转换为换行符，将语料库内所有大写字母转为小写字母，将处理后的语料库进行排序三步操作将语料库进行了分词处理并存入新文本。

语料库中多数单词有不同的形式，我们对分词处理后的语料库文件进行词形还原。安装TreeTagger包，使用tree-tagger-english命令对文件进行了词形还原并存入新文本以备用，针对TTR的比较时，我们认为词形的不同也能体现用词的丰富度，因此未作出词形还原处理。

在词汇层面处理完成后，为了解每个语料库之间的句子长度差异，我们对语料库原文件做了分句处理，原理同分词处理。通过tr命令，将语料库原文文件中的分号“;”和句点“.”作为分句标准 [4]，使它们转变为Linux系统可识别的“\ n”，存入新文本。

5 主要的生物信息学实验和实验结论

5.1 语料库间TTR的比较

在Linux中，从语料库整体的TTR入手，通过wc命令分别查看分词处理后文件的总词数，得到严谨型语料库总词数为854816，科普型语料库总词数为761228，AIMED语料库总词数为49168，GENEREG语料库总词数为77297，IEPA语料库总词数为14925，将其分别作为各自TTR计算的分母；通过sort命令-u参数的处理后，使用wc查看去重后的总词数并将其作为各自TTR计算的分子，得到严谨型语料库总词数为31849，科普型语料库总词数为32149，AIMED语料库总词数为6407，GENEREG语料库总词数为7862，IEPA语料库总词数为2998，计算各自的比值，获得了5个语料库整体的TTR值。为了减少实验误差以及增加结果的可信度，需要再对5个语料库进行随机抽样后假设检验，后比对5个语料库之间的TTR差异。

在R中，载入dplyr包，通过sampel函数对导入的文件进行随机抽样，每个均抽取5000组，记录每次抽取到的总词数和通过distinct函数去重后的总词数便于后续计算随机抽样得到的TTR值，由于每次抽取的词数由每个语料库的总词数决定，为了使抽取到的TTR值更接近整体TTR值从而提高准确度，我们每次抽取的词数为每个语料库总词数的85%。之后对每组5000个抽样的TTR值进行了正态检验，发现只有IEPA语料库的抽样是不符合正态性的，而其它四个语料库的抽样结果都有不同程度的符合正态性，这是不合理的，我们抽取出的TTR结果应比较平稳，不会不符合正态分布，所以通过查询资料得到了结果，我们找到了一可以说明这个现象存在的例子，并且这个例子的作者还简述了这个现象说明我们不能拒绝样本来自于正态总体的原假设：“我们取有序的正整数序列[1:30]进行Shapiro正态性检验，众所周知，正整数序列完全不是正态的”，而我们看到的结果是p值为0.2662，符合正态分布，作者又说“我们看到正整数序列[1:30]不能拒绝原假设，但它绝不是正态的，可以看到，p值大于0.05的显著性水平，但是我们不能证明样本数据就是服从正态分布的，只能说不能拒绝样本来自于正态总体的原假设”，所以我们抽样的TTR结果符合正态分布可能是因为这些数据能够存在于某个正态总体。在语料库之间执行var.test()函数，得到它们之间方差不同质的结果。根据正态性检验和方差同质性检验的结果，我们调整t.test()的参数，对每组语料库之间进行了t检验，发现它们之间均存在极显著差异，绘制更直观的箱型图，展现了每个语料库抽样TTR的差异（图1 a.）。 计算每个语料库抽样TTR的平均值，得到如下结果：

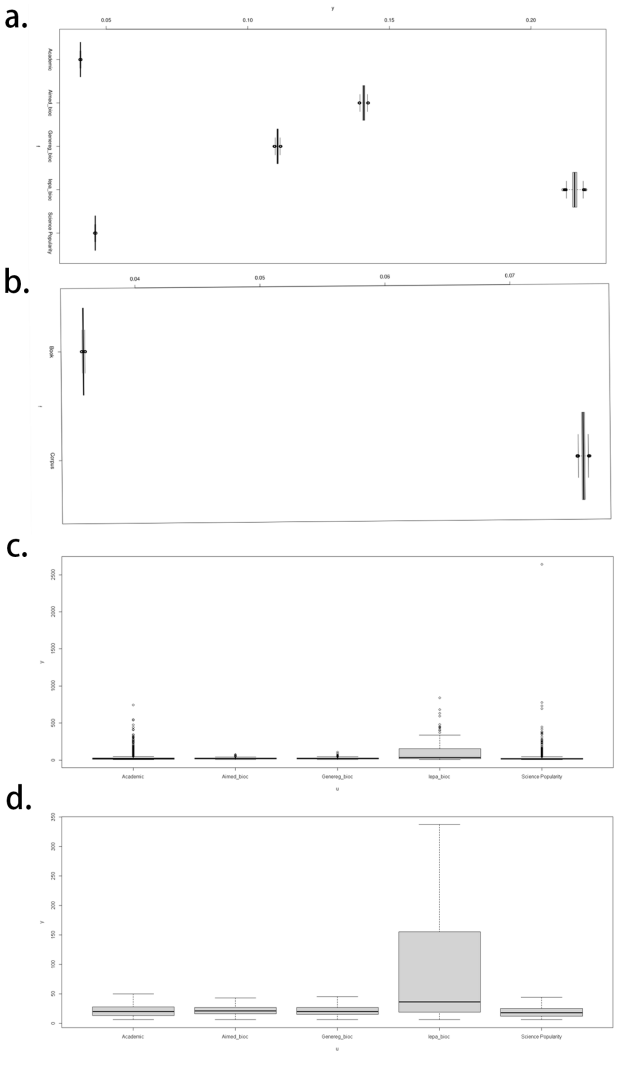


图1:
a.体现我们所使用的5个语料库的TTR差异的箱型图
b.为了进一步验证书籍和文献等的用词差异而作的我们使用的6本书籍与其它3个语料库的TTR差异的箱型图
c.我们此次所使用的5个语料库句子长度差异可视化的箱型图（未去离群值）
d.我们此次所使用的5个语料库句子长度差异可视化的箱型图（已去离群值）

图 1: 箱型图。图片来源:自绘，工具：R-4.0.5，Photoshop CC 2019。

严谨型书籍语料库的整体TTR为0.0372583，抽样获得的平均TTR为0.040917；
科普型书籍语料库的整体TTR为0.0422331，抽样获得的平均TTR为0.04614521；
AIMED语料库的整体TTR为0.1303083，抽样获得的平均TTR为0.1409662；
GENEREG语料库的整体TTR为0.1017116，抽样获得的平均TTR为0.1105369；
IEPA语料库的整体TTR为0.2008710，抽样获得的平均TTR为0.2154706。

抽样获取的平均TTR比整体TTR都要稍大一些，但语料库之间的差异关系依旧没有改变，并且在进行语料库的TTR分析时，我们发现词数较少的三个语料库的TTR均显著大于由书籍构成的字数较多两个语料库，所以我们不禁猜测由于用词的限制，导致当词量达到一定程度后，将出现很多的重复词汇，导致了TTR的下降，因此现在看起来由书籍构成的语料库的用词丰富度更低，为了验证这种猜想，我们将六本书分开，每本书作为一个语料库进行TTR的计算。我们使用同样的抽样方法进行抽样计算TTR，通过计算得到如下结果：

《GENEX》的整体TTR为0.0399536，抽样获得的平均TTR为0.04384342；
《物种起源》的整体TTR为0.0526115，抽样获得的平均TTR为0.05783426；
《生命是什么》的整体TTR为0.1124671，抽样获得的平均TTR为0.122679；
《昆虫记》的整体TTR为0.0306737，抽样获得的平均TTR为0.03380502；
《自私的基因》的整体TTR为0.0872458，抽样获得的平均TTR为0.09440558；
《昆虫记忆》的整体TTR为0.1093851，抽样获得的平均TTR为0.1186744。

同之前合并的TTR数据比对发现，每本书籍的平均TTR比合并TTR略高一点，并且六本书籍分开作为语料库后TTR确实比合在一起时的TTR明显要更高，我们基本验证了我们猜想是成立的，因此我们认为：语料库的TTR也会与语料库总词量相关，当语料库总词量达到一定程度后会受限于词汇的使用而出现大量重复词汇，导致TTR下降。我们进一步思考：语料库的TTR也能从一定程度体现了语料库的好坏，在保证词汇的正常使用和多样性的同时，构建语料库不宜采用过多的词量，对语料库的后续操作效果也会较好。这也是我们自己构建语料库的一个弊端，书籍构建的语料库词量过大，但词汇的丰富度又远远不足，导致了我们构建的两个语料库TTR较低的现象。

为了更清楚直观地感受生物类书籍语料库与生物医学语料库之间的差异，我们又用同样的方法，将上述6本书以及3个语料库分别作为语料库进行了对比，绘制了一张结果较为鲜明的箱型图（图1 b.），可以得知单一生物类书籍与生物医学语料库的用词差异也是非常显著的。

5.2 词云的绘制

在R中导入词形还原后的语料库文件，通过课上老师讲解的代码结合课后网上的教程，我们载入tm、SnowballC、wordcloud、RColorBrewer包进行后续实验。

对读入的文件使用tm_map()函数，调整参数删除特殊符号，英语停用词，多余空格等操作后，通过函数TermDocumentMatrix()和as.matrix()生成了文档矩阵后，对矩阵内的数据进行了降序排序，生成了词频数据框，最后使用wordcloud()函数进行了词云绘制，得到了五个语料库的词云图（图2）。

只是词云并不能让我们看出每个语料库中详细的词频差异，因此我们将每个语料库的词频数据进行了可视化，得到了5个语料库的词频分布图，可以看出语料库使用次数较多的词几乎集中在前两个词汇，AIMED语料库词频较高的词最多，有5个词在该语料库中出现频率较显著，对于我们研究的生物医学类语料库，除去较显著的词后，剩余词的使用频率都相差不大，而书籍里出现频率较显著的词更少，只有一两个，几乎可以断定一个语料库使用较多的词与该语料库的中心有关。

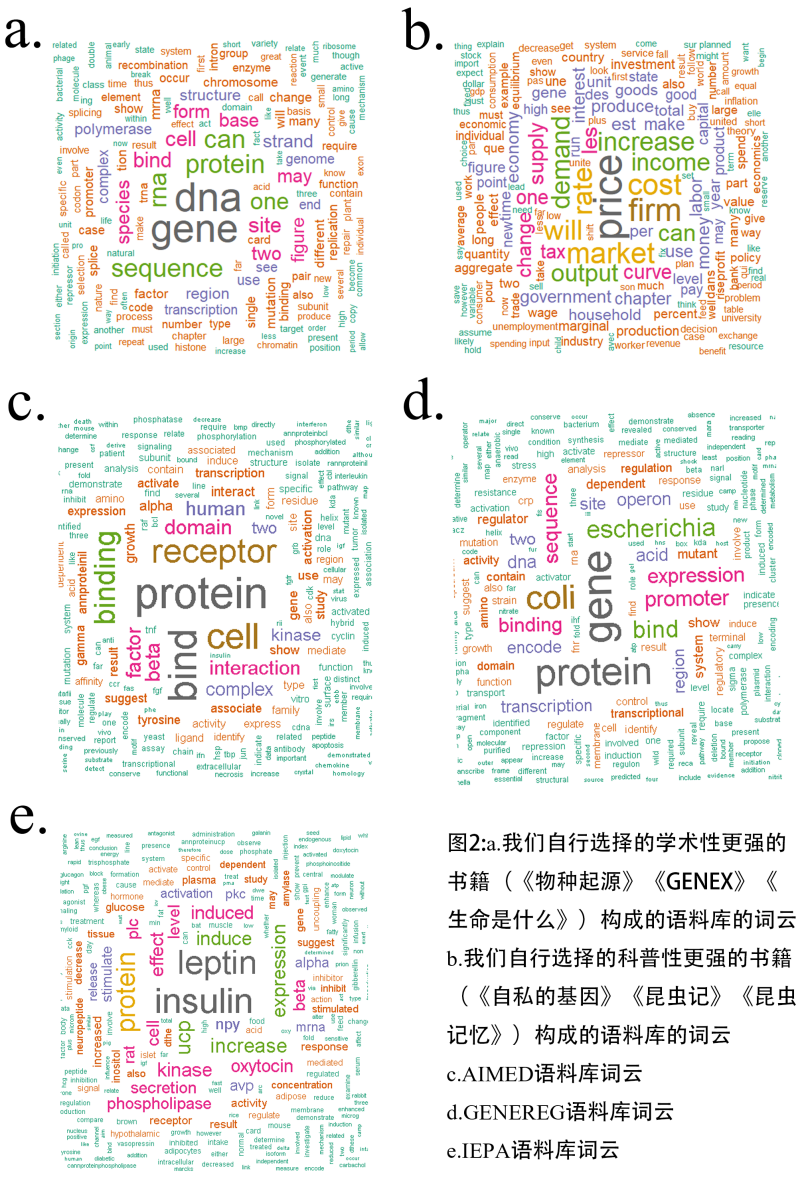


图2:a.我们自行选择的学术性更强的书籍（《物种起源》《GENEX》《生命是什么》）构成的语料库的词云
b.我们自行选择的科普性更强的书籍（《自私的基因》《昆虫记》《昆虫记忆》）构成的语料库的词云
c.AIMED语料库词云
d.GENEREG语料库词云
e.IEPA语料库词云

图 2: 词云。图片来源:自绘，工具：R-4.0.5，Photoshop CC 2019。

5.3 语料库句子长度分析

编写python代码，把分句处理后的文本导入后按行输出每句话的词数，作为5个语料库句子长度分析比较差异的基础数据，数据导入R中后，去除数据中数值小于等于5的数据，对剩下的数据进行不同语料库之间句子长度差异的分析。提取之后各语料库所剩行数如下：严谨型：34470；科普型：35133；AIMED：2011；GENEREG：3356；IEPA：113。

对每个语料库的句子长度数据进行抽样，我们查看了每个语料库的数据量，即筛选后所剩的句数，考虑到每次抽取的数据的数量相同以及数据抽少了对每个语料库的有效性不同，所以我们决定按照最大数量的语料库句数作为抽取数的15%并作小幅度提升，使用每个语料库抽取250000次，每次抽取的数量均为1的方法进行sample随机抽样，然后将数据分别存入事先准备好的空数据框进行后续分析。由于抽取的数量较多，此处耗时较长。

将抽样好的数据先进行一次正态性检验，发现5个语料库的句子长度都极不符合正态性分布，又对其进行了F测验，查看它们之间的方差同质性，检验后的p值均小于2.2e-16，证实这些语料库的句子长度两两

之间方差均不同质，所以可以进一步进行t.test()，并将参数修改为“paired=FALSE，var.equal=F”，然后来检验它们之间是否存在显著差异。结果显示，数据两两之间拥有极显著差异，由于p值小到一定程度后t.test()不会显示具体p值，无法直观了解它们之间的差异，所以我们通过作图将结果更直观地展现出来。

直接绘制箱型图会因为有大量的离群值（图1 c.），导致绘制的结果并不好，因此我们尝试对离群值进行处理。通过在boxplot()中添加参数“outline = FALSE”，在绘制箱型图时去除离群值（图1 d.）。对比两次箱型图，除了IEPA语料库的句子长度数值分布较广泛以外，其他4个语料库的结果差异并不显著，所以我们在这里初步猜测之前t.test()的结果所表明的所有语料库两两之间差异极显著很有可能是由大量的离群值所导致的。观察去除离群值前的箱型图，我们能够明显看出IEPA语料库句子长度的分布最广，离群值偏多，AIMED语料库和GENEREG语料库的离群值较少，分布较为集中，自行构建的严谨型语料库和科普型语料库有较多离群值，并且科普型语料库的句子长度离群值波动很大，这些从一定程度上支持着我们对“t.test()结果与箱型图直观感受不太一样的原因”的猜测。因此我们通过箱型图识别的方法去除了原数据中的所有离群值，然后又通过相同的方法对每组数据之间进行了一次t检验。

使用去除了异常值的数据进行t测验后，结果依旧表明5个语料库两两之间存在极显著差异，那么上述猜测就被推翻了，并不是由大量离群值造成的，于是我们猜测可能是由于IEPA语料库句子长度数值分布较广，所以在图像上展现出的结果显得其余4个语料库分布都较为集中，从而让我们觉得差异不显著，但这个猜测我们暂时没有想出办法去验证。

6 后记

6.1 课程论文GitHub网址

`https://github.com/Bing-nai/ziran_code.git/` 课程论文代码地址

6.2 课程论文构思和撰写过程

6.2.1 课程论文构思

我们对NLP是完全陌生的，在此次撰写课程论文中，我们秉承着以完成课上学到的知识为基础，尽我们所能进行拓展与开放思考。为了能够对NLP有基本的认识，实践前我们上网搜索关于NLP的发展历史，基本步骤，未来前景等，在了解到NLP的基本步骤后，我们意识到，此次所进行的实验只是NLP学习最基础的一步，未来对于NLP的深入学习，我们还有远远的路要走。

在语料库的选择上，我们最开始是想选择5个生物医学类语料库的，但仅仅选择生物医学类语料库对于结果的分析可能会很单一，且我们注意到目前BioNLP权威会议把事件抽取作为了主导任务，采用生物医学文献为数据源，以支持开发更细粒度，更具结构化的数据库为目的，引导人们提出各种生物事件抽取方法 [5]。所以我们在语料库的选择上采取了“3+2”，即3生物医药类语料库+2我们自己构建的生物类书籍语料库。通过文献阅读，我们了解了IEPA语料库与AIMED语料库在NLP类文献内使用频率较多，所以我们选择了这两个出现频率较高的语料库以及出现频率较低的GENEREG语料库 [6]。

确定了语料库后，我们就开始思考该如何分析语料库，除了课上所学的TTR与词云绘制，我们还可以做什么？在查阅一些资料后，发现可以分析的东西还是很多的。例如词形还原，词性标注，去停用词等。由于此次选择的语料库与情感分析，知识推理无关，所以我们在进行词性标注后发现无法继续分析便舍弃了。

具体操作过程中，有时会出现与预料不同的结果，我们会尝试对其进行扩展与发散思考，做出猜测并进行检验。

6.2.2 课程论文撰写

由于对语料库的操作及得出结果都是直接得到的，所以我们在撰写过程中就思考是否将每次操作后的结论单独提取出一个板块，但简单的尝试后我们发现有很多操作结果与结论可以穿插进行，单独提取出会导致实验流程部分语意不通且结论部分无序零碎。纠结过后，我们还是选择操作结果与结论穿插撰写。

撰写论文前出现了一点小插曲，在TexWorks中，我们尝试将文件保存为PDF格式时，会有latexpdf配置错误的提示，通过查询一些之前人们的解决方法，下载texlive解决了这个问题。

在撰写论文代码的过程中，出现的最大的问题就是参考文献的导入，Bibtex格式引入参考文献时，引用的数字符号会以问号形式呈现，且最终的参考文献处无任何显示，尝试了多种方法后，终于通过命令提示符的bibtex 文件名称.aux命令得以解决；在复查修正中，我们发现了更见简洁的导入参考文献的方式，也算是很大的收获。

6.3 所参考主要资源

`http://corpora.informatik.hu-berlin.de/` 部分生物医学类语料库下载网址

`https://genialebooks.com/ebooks/` 生物类书籍下载网址

`http://www.sthda.com/english/wiki/text-mining-and-word-cloud-fundamentals-in-r-5-simple-steps-you-should-know/` 词云绘制参考代码网址

6.4 代码撰写的构思和体会

在词云的绘制中，因为使用TreeTagger包进行词形还原时有些词未能识别而显示“unknown”结果，所以我们决定将所有“unknown”均替换为对应位置的原词，调整后重新读入R中，继续词云的绘制。

在语料库句子长度比较中，我们对数据进行t检验时，发现Shapiro.test()函数只能检验数量在3到5000之间的数据，所以我们改变思路，采取了与之类似的，nortest包里的Anderson-Darling进行正态性检验，并且还把每组数据的前5000个数据进行Shapiro.test()来与之对比，发现其实两种方法在此次实验中并无差异。

6.5 生物信息学实验设计的构思和体会

生物信息学是一门交叉学课，在我们掌握生物知识的基础上，我们还需要熟练掌握代码的编写来适应飞速增加的生物学数据。每一次的代码编写都是一场实验，所以如何设计实验让代码完美是值得思考的，我们应在正式实验前充分了解实验的背景和前人做过的尝试，减少自己的弯路，要大胆进行尝试，在反复的试错中也许会有新的发现或者新的想法，新的发现和想法要尽可能验证，不断的推翻与建立，我们才会学到更多的知识，逐渐接近正确的方向。

6.6 人员分工

为了达成人均得到练习的目的，我们采取了双线程并行的方式，共同完成了实验数据预处理、词云绘制、TTR分析、句子长度分析、文献查找、论文撰写等一系列工作。其中，实验数据预处理由二人对半完成，TTR分析，词云绘制，句子长度分析由二人对半完成大部分工作，在最后处理上，杨瀚轶负责TTR与句子长度分析的整合和比对，王世松负责词云的整合和比对以及对句子长度分析的复查，文献查找由二人共同商讨搜集，论文撰写部分由杨瀚轶完成第5部分与人员分工部分的撰写，其余部分由王世松完成，最终复查由二人共同完成。

参考文献

[1] mantch. 自然语言处理（nlp）的发展. https://www.cnblogs.com/mantch/p/11385113.html/.

[2] Kevin陶民泽. 自然语言处理的一般流程. https://baijiahao.baidu.com/s?id=1651906999137532506/.

[3] G McKee, D Malvern, and B Richards. Measuring vocabulary diversity using dedicated software. In *Literary and Linguistic Computing*, pages 323–328, 2000-9.

[4] 刘敏捷. 基于组合学习和主动学习的蛋白质关系提取. PhD thesis, 大连理工大学, 2015-6-8.

[5] 王健. 面向生物医学领域的信息提取的关键技术研究. PhD thesis, 大连理工大学, 2014-5-16.

[6] 郭瑞. 基于迁移学习和词表示的蛋白质交互关系抽取. Master's thesis, 大连理工大学, 2015-5-5.

3.3 聂有攀《生物医药语料库词汇分析》

语料库通俗来讲是某些特定文本的集合。字典，是最为常见也最为我们熟知的一种语料库。语言学家希望通过对语料库进行分析从而得到这些特定文本的规则与联系。本文以 GENIA 和 ACGC 两个语料库为例，阐述了如何下载获取这两个语料库，并使用 python 代码围绕这两个语料库的 TTR 值探寻这两个语料库的异同。

生物医药语料库词汇分析

聂有攀 [1]

[1]华中农业大学信息学院，430070，武汉，湖北，中国

摘要

语料库通俗来讲是某些特定文本的集合。字典，是最为常见也最为我们熟知的一种语料库。语言学家希望通过对语料库进行分析从而得到这些特定文本的规则与联系。本文以 GENIA 和 ACGC 两个语料库为例，阐述了如何下载获取这两个语料库，并使用 python 代码围绕这两个语料库的 TTR 值探寻这两个语料库的异同。

关键词: ACGC,GENIA,corpus, TTR

1 课题概况

本学期开设的这门课程是一门很实用的学科，我希望能在课程的学习中学到有关生信专业的知识。本论文选题难度低，容易完成，以 GENIA和 ACGC两个语料库为例，阐述了如何下载获取这两个语料库，并使用 python代码围绕这两个语料库的 TTR值探寻这两个语料库的异同

2 语料库下载

该部分中包含两个目标语料库的相关简介以及如何从公共网络资源中搜索并下载语料库资源。

2.1 GENIA语料库

GENIA语料库由 GENIA项目创建并添加注释。该语料库的建立旨在支持通过文本挖掘提取文献中的生物学信息。该项目研究者通过 Pubmed检索人类、血细胞和转录因子这三个术语，筛选出 2000篇 Medline数据库内文章的摘要，这些摘要的集合即为 GENIA语料库，其中包含超过 40万个单词和近 10万个针对生物学术语进行手工编码的注释。[1]在这里给出下载 GENIA语料库的方法。

如果觉得不想通过别人的文章获取数据，希望能直接下载官方发布的语料库，获取一手资料，那么我们可以找寻 GENIA的官网。通过百度搜索“ GENIA”，第一个即为 GENIA项目的官网。 home界面为GENIA项目的一些简介。通过点击左侧的目录栏可以获取更多想要的信息。如果想要获取与第一种方法得到的一样的压缩包的话，在 Part-of-speech annotation下即可下载。官网同样还有其他格式的 GENIA语料库可供下载，在此不在赘述，后续代码处理的内容文中方式获取的语料库为准。

地址. http://www.geniaproject.org

2.2 AGAC语料库

AGAC语料库是一个关注于生物医学文章中功能获得性（ GOF）或功能丧失性（ LOF）的突变基因的语料库。该语料库包含 1000篇从 pubmed中筛选出的文章的摘要。[2]此处同样给出方法下载获得 AGAC语料库数据。 百度很难搜到相关的信息，于是我选择用谷歌搜索。搜索关键词 “AGAC corpus”后，“AGAC

Track -Google Sites”，该网站即为 AGAC的官方网站。在右侧时间或底部“ Data and Evaluation Codes”处选择相应超链接即可下载语料库。

地址. https://sites.google.com/view/bionlp-ost19-agac-track

3 TTR计算

3.1 TTR简介

TTR的全名为 Type-Token Ratio，即形符与类符之比，通俗来讲就是出现的词的种数除去词的总数的值。这一数值可以反映所统计的文本的复杂程度。以一篇英文文章举例，如果文章的作者选用了很多不同的单词进行造句，那么 TTR值就会接近于 1，说明这篇文章用词越为复杂，对于语料库来说亦是如此。本次任务中我们从统计 TTR的角度来分析 GENIA语料库与 AGAC语料库之间的异同。

$$TTR = \frac{Amountofuniquewords}{Amountoftotalwords} \tag{1}$$

3.2 数据预处理

GENIA 语料库下载的格式为压缩包，解压后得到语料库的 xml 文件，由于本次实验主要目的为获取文本信息。因此通过浏览器将 xml 文件打开，全选后将所有文字复制粘贴到 txt 文件中方便之后的代码处理，之后使用代码直接读取文本即可。AGAC 语料库下载并解压后是文件夹，摘要为 txt 文件储存在文件夹内，需要批量读入目录下的所有文件。

3.3 Linux 代码实现

该部分为夏老师在课上提供的代码，具体思路为，读入文件后将非字母和数字的替换为换行符，换句话说符号和空格都替换为了换行符。这样文件每一行都是一个单词，并将所有大写字母转换为小写。之后用 sort 命令排序后用 uniq 去除重复单词。用 wc 命令统计单词数。根据公式 1 即可计算出 TTR 值。下为将非字母和数字的替换为换行符并将所有大写字母转换为小写的代码。

```
cat AGAC. txt | tr -cs " [ : alnum : ] " "\n" | tr [ : upper : ] [ : lower : ] > AGAC. \_word . txt
```

结果为 genia 的 TTR 为 0.0334，AGAC 为 0.1214。

3.4 python 代码实现

python 代码目的是对一些后续内容能更方便地使用调包实现。python 代码选择手动筛选符号替换为空格，再以空格为分割符切割出各个单词的方式。下为将符号替换为空格的代码。

```
#去掉标准符号（ 可在punctuation 中添加想去掉的标点， 直到符合预期）
punctuation = ' ]! ,.:%&+* > <=:/?"\ '[}{()- '
nty=re.sub(r'[{}]+ '.format(punctuation),'␣',gtry)
```

结果输出:

genia_TTR= 0.03338843882551405，agac_TTR= 0.12233719832816503

与 linux 计算得出的结果相比其实差距并不是很大。

3.5 STTR 简介与计算

STTR，也可称为标准 TTR（Standard TTR），定义为计算每 n个词的 TTR后再取其平均值，n值可以根据语料库规模调整取值。公式 2为 STTR计算公式参考。j为语料库根据 n值分割所得的组数。

$$STTR = \sum_{i=1}^{j} STTRi/j \tag{2}$$

先计算各个分隔区间的取段计算 TTR 后，然后取平均值。下为计算 agac 的 STTR 代码。

```
# 计算TTR函数
def t t r (m) :
    x=len (m)
    y=list(set(m))
    y=len(y)
    return f l o a t ( y/x)

ts=int(1200)    #设置取词间隔为1200
times1=int(len(a_token)/ts)
sa =[]
for i in range(times1):
    i0=i*ts
    i 1 =( i +1)* ts
    sa . append ( t t r ( a_token [ i 0 : i 1 ] ) )
```

结果：a_STTR(1200)=0.432517 g_STTR(1200)=0.379243

3.6 结论

GENIA语料库的 TTR值相比与 AGAC语料库来说小了太多。这一点可以从语料库的规模来进行解释，GENIA语料库选用了 2000篇的摘要，而作为对比本次实验中只选用了 AGAC中包含 250篇摘要的训练数据。可以预想文本量越大则单词的重复率会越高，对应于公式 1的情况就是分母不断增加，而分子的增加很少，则 TTR值就很小。因此应关注 STTR这一经过标准化的参数。那么这里得出的结果是 AGAC语料库的 STTR值比 GENIA语料库更高，由此可以提出推论： AGAC语料库可能比GENIA语料库复杂程度更高。

4 后记

通过学习 “自然语言处理与知识发现”这门课。为我带来了一种全新的关于如何分析出有价值生物信息的方法。同时得益于各位同学的展示，让我在课程之外可以看到更多的思想与方法。

4.1 课程论文构思和撰写过程

本文的构思是取自于老师课堂上同学的分享和一部分的网络资料，撰写该文时请教了其他同学，同时从中学习到了论文的撰写规范。

4.2 代码撰写的构思和体会

通过这段时间的课程和代码的练习，使我对这门课程的性质、基本理念、设计思路、课程目标有了全面的了解，明确了新课程学习的目标和方向。但是如果完全适应未来的学习和应用，不论是从知识上还是学习方法上都需要不断地完善，希望能在未来的学习里不断变强。

参考文献

[1] Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. GENIA corpus - a semantically anno- tated corpus for bio-textmining. In *Proceedings of the Eleventh International Conference on Intelligent Systems for Molecular Biology, June 29 - July 3, 2003, Brisbane, Australia*, pages 180–182, 2003.

[2] Yuxing Wang, Kaiyin Zhou, Mina Gachloo, and Jingbo Xia. An overview of the active gene annotation corpus and the bionlp ost 2019 agac track tasks. In *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks*, pages 62–71, 2019.

4

基于人类表型本体 HPO 的富集分析

我们通常是从 *GO* 了解本体及其注释的，那么何为富集？

– Jingbo Xia

项目要求：

自由选择 GO 富集，或者 HPO 富集的课题。若选用 GO 富集，可选用 TopGO 的 R 包，如果选用 HPO 富集，可选用 HPO 富集的往年参考代码，也可以自编。

提示：请仔细揣摩课堂讲授的富集分析理论。相关代码采用 R 代码较为容易实现。

参考的刘彤师兄 HPO 富集项目：`https://github.com/tongliu-liu/HPO-enrichment-analysis`

参考的 GO 富集项目：`https://github.com/bionlp-hzau/Tutorial_4_GO_Enrichment`

相关论文三篇：

晏倫一《HPO 富集分析》

杨晓龙、周旺《HPO 富集分析》

杨迟、屈伸洋《HPO 富集分析》

4.1 晏倫一《HPO 富集分析》

艾滋病是一种危害性极大的传染病，由感染艾滋病病毒 (HIV) 引起。HIV 是一种能攻击人体免疫系统的病毒。它把人体免疫系统中最重要的 CD4T 淋巴细胞作为主要攻击目标，大量破坏该细胞，使人体丧失免疫功能。因此，人体易于感染各种疾病，并可导致恶性肿瘤，病死率较高。本次实验基于 HPO 富集分析，对文献中给出的数据进行富集分析，对 HIV 攻击人体免疫细胞出现的疾病进行汇总分析，对 HIV 进一步了解。

课程论文 GitHub 网址：`https://github.com/lunyiy/bioNLP/tree/main/Course-project`

HPO 富集分析

晏倫一 [1]

[1] 华中农业大学信息学院，430070，武汉，湖北，中国

摘要

艾滋病是一种危害性极大的传染病，由感染艾滋病病毒（HIV）引起。HIV 是一种能攻击人体免疫系统的病毒。它把人体免疫系统中最重要的 CD4T 淋巴细胞作为主要攻击目标，大量破坏该细胞，使人体丧失免疫功能。因此，人体易于感染各种疾病，并可导致恶性肿瘤，病死率较高。本次实验基于 HPO 富集分析，对文献中给出的数据进行富集分析，对 HIV 攻击人体免疫细胞出现的疾病进行汇总分析，对 HIV 进一步了解。

关键词：HIV，免疫细胞，HPO，疾病

1 课题概况

首先感谢祝宏涛同学的选课推荐，以前不曾接触过自然语言，认为自然语言是一门计算机语言，所以选这门课也是想更多的了解一下课程实际内容。选择以 HIV 感染人体免疫细胞 HPO 富集分析为课题也是为了更多地了解艾滋病。社会上的一些事情总是导致人心惶惶，比如共享单车坐垫下、网吧座椅下暗藏的针头，一经报道便容易令人猜测是否是带有艾滋病病毒。计划以此作为一个切入点，来了解并找出有关艾滋病的一些发病症状及发病原因。

2 数据

此课程项目的基因集选自 NCBI 网站有关 HIV 的论文文献：Chronic HIV infection induces transcriptional and functional reprogramming of innate immune cells[1]。

背景数据集来源于 HPO 官网最新版本的数据，数据链接：`http://purl.obolibrary.org/obo/hp/hpoa/genes_to_phenotype.txt`，用于差异化基因 GO 分析可视化的数据是经过 GO 分析网站 DAVID 直接得到的，网站链接：`https://david.abcc.ncifcrf.gov/summary.jsp`

3 研究方法

3.1 研究方法的算法背景，与其他方法的联系与区别

本次项目所用富集分析方法为传统富集分析即 ORA (Over-Representation Analysis), 同时进行简单的 GO 富集分析, 将得到差异化基因的功能相关图进行结合分析。其他可用于富集分析的方法有 PT (pathway topology)、NT（network topology）和 GSEA（Gene Set Enrichment Analysis)。

(p 值计算方法) 超几何分布：

$$P_Q = \frac{C_M^Q * C_{N-M}^{K-Q}}{C_N^K} \tag{1}$$

N：HPO 背景数据集中所有 gene 数目; M：HPO 背景数据集中某个 Term 的基因数目；K：差异表达基因数目；Q：该 Term 在差异基因中对应的基因数目;

FDR 校正

$$q_{value} = \frac{p_{value} * n}{rank} \tag{2}$$

p_{value}：超几何分布方法计算得到 p 值; n：超几何分布方法得到的数据量（p_{value} 数）; rank：p_{value} 从小到大排序后的次序

FDR 校正的目的就是控制 q 值通过多重检验校正减少假阳性发生次数。在 R 中可以使用 fdrtool 包对 P 值进行 FDR 校正，得到 q 值，用 q 值对数据进行筛选。

3.2 研究方法中的核心思路

本文旨在对 HIV 攻击人体免疫细胞后的差异化基因进行 HPO 富集，同时利用现成的网站进行简单的 GO 富集并进行可视化, 通过与临床症状的比较进行对比分析，对 HIV 进行进一步了解。

3.3 本文的方法部分与课堂讲授内容的联系和区别/或补充

本文的方法采用传统的富集分析流程：获取差异基因、富集分析以及结果分析，在富集分析中采用的是超几何分布计算 p 值，然后使用 FDR 校正得到 q 值，通过 q 值来筛选 HPO 子集，降低了在 p 值计算时的误差，使富集结果更加可靠。

4 算法实践和代码编写要求

4.1 任务描述

(1) 根据研究方法对数据集进行 p 值计算，再利用 R 包 fdrtool 对 p 值进行校正，得到 q 值，选取 q 值小于 0.05 的 HPO 子集，使用 R 包 ggplot 可视化。

(2) 通过 DAVID 网站分析现有的差异化表达基因，得到 GO 富集结果，使用 R 包 ggplot2 进行可视化。

4.2 实验设计

(1) 软硬件条件

window 10; R: version 4.0.5

(2) 数据集划分

因为原始数据类型已是差异表达基因，只需根据描述分成上调基因集和下调基因集

(3) 数据预处理

对 HPO 官网的背景数据集进行过滤，仅保留 5 列，分别为 entrez-gene-id、entrez-gene-symbol、HPO-Term-ID、HPO-Term-Name、disease-ID for link。

(4) 算法实施

分别对基因集计算 p 值，然后进行 FDR 校正，再筛选 q 值小于 0.05 的 HPO 子集使用 ggplot 可视化

4.3 某些关键代码

```
代码来源: https://github.com/tongliu-liu/HPO-enrichment-analysis/blob/master/HPO-enrichment-analysis.R
#p值计算
pValue2 <- data.frame(pValue = character())
for(valueId in hpoId) {
    pgeneInCategory <- sumData[sumData$hpoId== valueId,]
    k <- pgeneInCategory$kValue
    M <- pgeneInCategory$mValue
    p <- phyper(k-1,M, N-M, n, lower.tail=FALSE)
    pValue <- c(p)
    pFrame <- data.frame(pValue)
    pValue2 <- rbind(pValue2,pFrame)
}
#筛选q值小于0.05的HPO子集
hpoData <- unique(hpoData[hpoData$pValue<0.05,])
#画图，已封装成函数，输入参数为保存为文件（对文件中的英文进行了注释）的HPO富集数据
HPO.FileToPlot <- function(hpoFileName = "HPO.csv") {
  if(substring(hpoFileName,nchar(hpoFileName)-3) != '.csv'){
    hpoFileName <- paste(c(hpoFileName, 'csv'), collapse = '.')
  }
  data <- read.csv(hpoFileName, header = TRUE)
  ggplot(data = data,aes(x =Description,
                         y = Count,
                         fill =-log10(pValue))) +
  geom_bar(stat="identity") +
  scale_x_discrete(limits=data$description) +
  coord_flip() + labs(title = "EnrichmentHPO") +
  theme(plot.title = element_text(size = 15,face = "bold"),
        axis.text = element_text(size = 8,face = "bold"),
        axis.title.x =element_text(size=14),
        axis.title.y=element_text(size=16),
        panel.background = element_rect(fill="white", colour='gray')) +
  scale_fill_gradient(low = 'blue', high = 'red')
}
```

```
代码来源: https://mp.weixin.qq.com/s?__biz=MzU5NjA2MzAwMA==&mid=2247483886&idx=1&sn=317
    eaf8f25f839353c1882a8b2c2fe83&chksm=
    fe692799c91eae8f6da54623a1d637588b42e875680eaf3b9f37d2d16f389576b3b65a912512&mpshare=1&scene=23&srcid
    =0427yuaUjgjHrgLjurm39vud&sharer_sharetime=1619497889815&sharer_shareid=
    e9bed67a9cd4327372f54d495427a218#rd
#GO分析 ggplot2可视化
p = ggbarplot(data = allGo,x = "ID",y = 'Count',
              fill = "Category",
              palette = c("cadetblue3","mediumslateblue","mediumorchid3"),
              sort.by.groups = T,xlab = '',ylab = "Target␣genes")
ggpar(p,x.text.angle = 90)
ggsave(plot = p,'barplot.pdf',width = 10,height = 4)
```

5 主要的生物信息学实验和实验结论

5.1 结果与分析

5.1.1 上调基因 HPO 富集结果分析

HIV 病毒感染人体免疫细胞上调差异表达基因 HPO 富集分析结果如图1所示。差异表达最显著的是复发性曲霉菌感染（Recurrent Aspergillus infections,HP:0002724），其次是口腔溃疡（Aphthous ulcer,HP:0032154）、盘状红斑皮疹（Discoid lupus rash,HP:0007417）、复发性口疮性口炎（Recurrent apht-

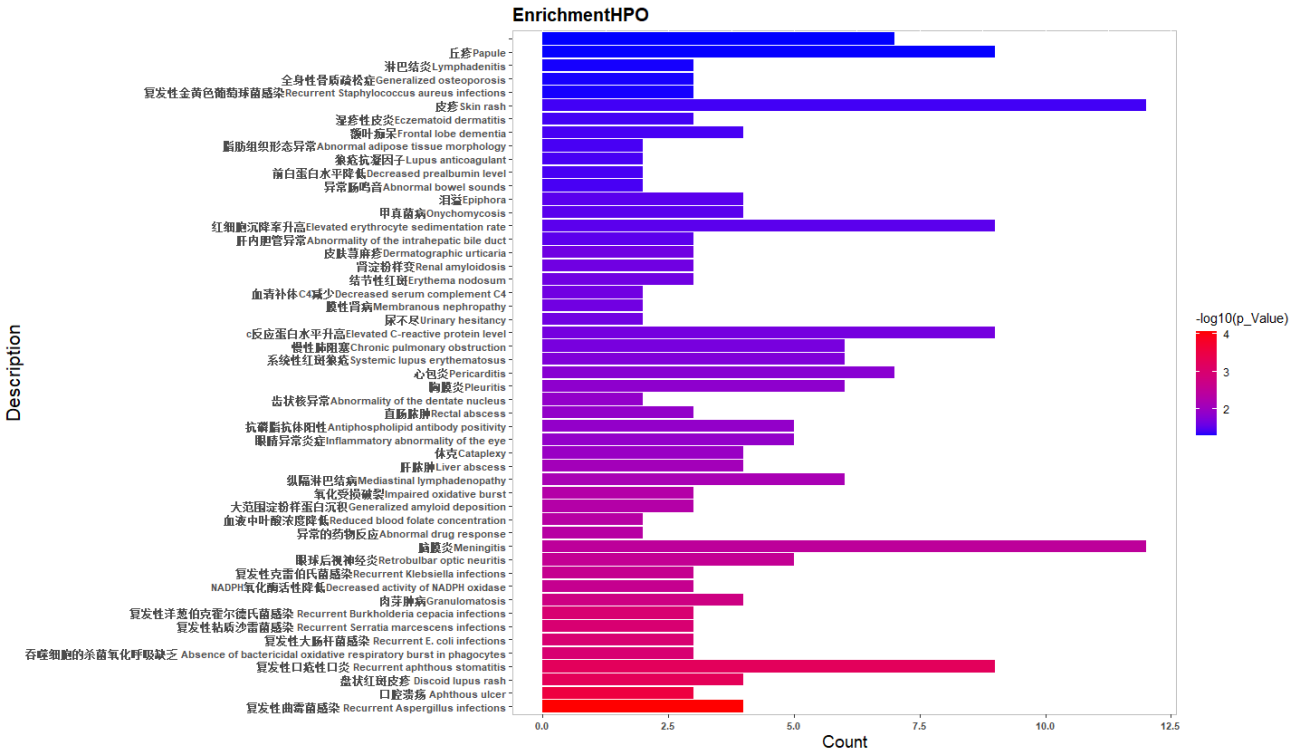


图 1: 上调差异表达基因 HPO 富集分析柱状图。图片来源：自绘；工具：R

hous stomatitis,HP:0011107)。富集最多基因的 HPO 子集是脑膜炎（Pleuritis,HP:0001287）和皮疹（Skin rash,HP:0000988)。在差异化基因最显著和拥有富集最多差异化基因的疾病中，皮肤类疾病占了三分之二。

同时，上调基因中主要的差异基因有 CYBB、NCF1、NCF2。CYBB 参与组成的细胞色素 b（-245）是吞噬细胞消灭微生物氧化酶系统的主要组成部分。查阅资料得知，NCF1、NCF2 和膜结合的细胞色素 b（558）是激活潜在 NADPH 氧化酶所必须的。富集基因主要与人体免疫相关，上调基因富集得到的疾病症状主要是皮肤病，富集基因结合富集得出的主要疾病症状，得到简单结论：HIV 攻击免疫细胞致使人体免疫功能下降甚至丧失，病原体更加容易侵入人体，出现各种皮肤症状。

5.1.2 下调基因 HPO 富集结果分析

HIV 病毒感染人体免疫细胞下调差异表达基因 HPO 富集分析结果如图2所示。差异表达最显著的是慢性溶血性贫血（Chronic hemolytic anemia,HP:0004870），其次是铁稳态异常（Abnormality of iron homeostasis,HP:0011031）、平均小体体积减小（Decreased mean corpuscular volume,HP:0025066）和低色小红细胞性贫血（Hypochromic microcytic anemia,HP:0004840)。富集最多基因的 HPO 子集是胆石病（Cholelithiasis,HP:0001801），其次是慢性溶血性贫血（Chronic hemolytic anemia,HP:0004870）、铁稳态异常（Abnormality of iron homeostasis,HP:0011031）、平均小体体积减小（Decreased mean corpuscular volume,HP:0025066）和低色小红细胞性贫血（Hypochromic microcytic anemia,HP:004840)、血凝过快（Hypercoagulability,HP:0100724)、髓外造血组织增生（Extramedullary hematopoiesis,HP:0001978)、慢性感染（Chronic infection,HP:0100724)、缺铁性贫血（Iron deficiency anemia,HP:0001891)、低纤维蛋白原血症（Hypofibrinogenemia,HP:0011900)。

下调基因中主要的差异基因是 HBB，HBB 基因是控制血红蛋白的基因之一，HBB 基因突变将会导致人体出现贫血症等相关疾病。结合疾病富集与差异化基因得到一个简单结论：HIV 攻击免疫细胞还会使人体编码血红蛋白的基因发生突变，导致人体患上贫血症等相关疾病。

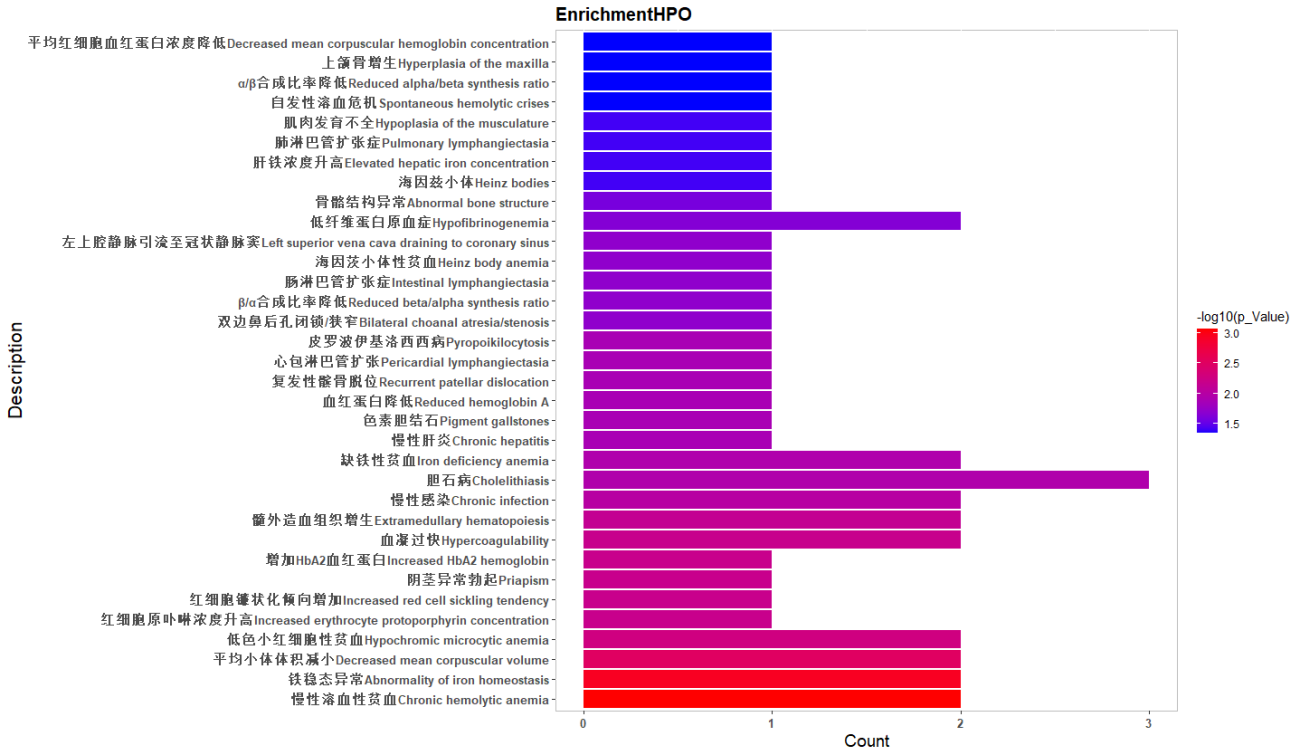


图 2: 下调差异表达基因 HPO 富集分析柱状图。图片来源：自绘；工具：R

5.1.3 差异化表达基因 GO 分析

HIV 病毒感染人体免疫细胞差异化表达基因 GO 分析如图3所示。从条形图中可以直观看出差异表达的基因跟细胞组分、分子功能和细胞过程都有很大的相关性，其中与细胞组分的相关性最大。可推测 HIV 攻击人体免疫细胞主要干扰人体免疫细胞的组分相关基因，导致细胞缺少部分结构，丧失部分甚至全部功能，继而影响整个人体免疫系统。在 GO 富集分析子集 GO:0005829 中含有 HPO 出现的主要基因 HBB、NCF1 和 NCF2 等，其余还富集了另外接近 70 种基因，由于时间关系，在此不详细说明。同时在细胞组分的相关性最高的 GO 子集 GO:0043235 中富集基因 NRP1、LRP1、KLRC2、TLR7。NRP1、KCLR2 均与肿瘤相关，TLR7 属于 TLR 家族，主要作用是识别病毒单链 RNA，介导抗病毒的天然免疫反应。通过已分析的基因可以猜测，HIV 入侵人体免细胞，破坏免疫细胞的相关细胞组分，使免疫细胞的 TLR 家族系列基因减少甚至停止表达，HIV 病毒能够顺利进行自我复制，同时破坏更多的免疫细胞，此举在人体中形成一个恶性循环。

5.2 实验结论

根据搜集的材料，HIV 感染后，一旦发展为艾滋病，病人就会出现各种临床表现。一般初期的症状如同普通感冒、流感样，全身疲劳无力、食欲减退、发热等，随着病情的加重，症状日渐增多，如皮肤、黏膜出现白念球菌感染，出现单纯疱疹、带状疱疹、紫斑、血疱、淤血斑等 (图1)；之后渐渐侵犯内脏器官，出现原因不明的持续性发热，可长达 3 ~ 4 个月；还可出现咳嗽、气促、呼吸困难、持续性腹泻、便血、肝脾肿大、并发恶性肿瘤等 (图2)。结合此次 HPO 富集分析结果 (图1、2) 和 GO 富集分析结果 (图3)，HIV 攻击免疫细胞，侵入 CD4T 淋巴细胞，破坏其细胞组分，导致免疫细胞的相关功能丧失，如识别病毒、产生免疫反应，同时，因为免疫功能的减弱或者丧失，不仅外来病原体能够轻易入侵人体，而且连人体自身的细胞病变也无法及时进行清除，这就表现在抵抗力极差，一些小病易得但不易治愈，皮肤病等反复发作，而且易患恶性肿瘤并发生癌症病变 [2]。同时因为癌细胞等抢夺营养，人体正常细胞营养不足、氧气不足，对身

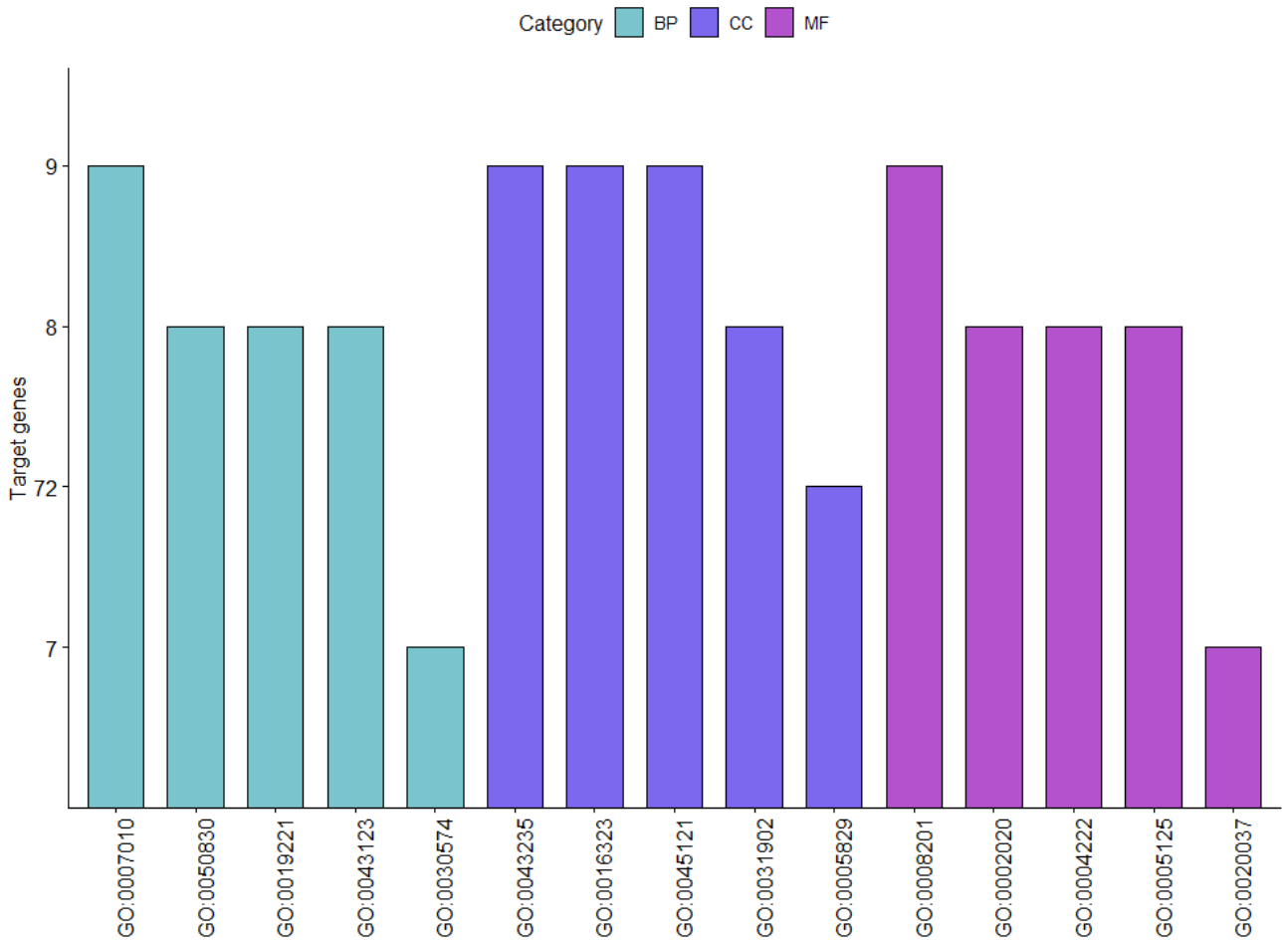


图 3: 差异化表达基因 GO 分析图。图片来源：自绘；工具：R

体供能不足，这就产生了全身疲劳无力、咳嗽、气促等相关临床症状。

6 后记

6.1 课程论文构思和撰写过程

主要构思是如何将所选课程项目完整流畅地展现出来，让别人读懂这篇文章，同时也对艾滋病有更深的了解，在撰写的过程中遇到了好多问题，例如 latex 的公式怎么用、参考文献怎么用，图片怎么引用，不过这些问题在朋友和百度、CSDN 的大佬们的帮助下顺利解决。

6.2 所参考主要资源

(1) 差异表达分析概念及代码参考：https://www.jianshu.com/p/b55276e46f0c

(2)FDR 校正原理参考：http://www.360doc.com/content/18/0914/21/19913717_786724085.shtml

(3)HPO 富集代码参考：https://github.com/tongliu-liu/HPO-enrichment-analysis/blob/master/HPO-enrichment-analysis.R

(4)HPO 富集代码参考：http://xiajingbo.weebly.com/uploads/1/3/3/0/13306375/enrich.pdf

(5)GO 分析及可视化代码参考：https://mp.weixin.qq.com/s?__biz=MzU5NjA2MzAwMA==&mid=2247483886&

idx=1&sn=317eaf8f25f839353c1882a8b2c2fe83&chksm=fe692799c91eae8f6da54623a1d637588b42e875680eaf3b9f3
mpshare=1&scene=23&srcid=0427yuaUjgjHrgLjurm39vud&sharer_sharetime=1619497889815&sharer_shareid=
e9bed67a9cd4327372f54d495427a218#rd

6.3 生物信息学实验设计的构思和体会

在最开始的时候其实并没有想好要做哪个方向的 HPO，于是我就从我身边的朋友下手，问他们怕什么病，第一个朋友就给了我这个答案：艾滋病，他跟我说，他老是看到哪个地方的共享单车坐垫里面出现针头，每次骑共享单车总是害怕单车座椅里面有针头，然后我联想到以前有经常去网吧的朋友跟我说，哪个网吧有个人被座椅里的针头扎了，说这是艾滋病人报复社会。其实当时这也是我的一个阴影，好在我不去网吧，影响不大，但是共享单车坐垫里面有针头是真的让我感觉到恐惧，所以我想更多地了解一下艾滋病，从生物文本挖掘的角度看艾滋病，了解艾滋病的发病症状，更好地了解艾滋病才能更好地预防艾滋病，于是我就选择了以 HIV 作为此次课程项目的方向。在选择数据的时候是比较轻松的，NCBI 有关 HIV 的文献不少，所以我就找了一个基因集相对较少的文献，然而在进行代码实践的时候我就遇到了好多问题，首先就是 R 包的版本不适配，在不断修改了 R 版本后，我的 R 软件不仅没有解决版本不适配的问题，还出现了 namespace 载入版本过低，无法导入 ggplot 包，于是我再度开启百度搜索之路，对 namespace 版本过低的 R 包进行卸载，但是问题仍然没有被解决，于是我暂时放弃了用 R 代码来写 HPO 富集分析。我投向了 python 的怀抱，开始 python 用得挺舒服的，但是一个关键包可以下载，但是死活无法调用的时候，我又烦恼了，我向一个朋友诉苦，他跟我说 HPO 富集使用 R 还是更简单一点，R 出现问题可以把 R 和 RStudio 完全卸载了试试，把数据删除干净，R 包版本不适配可以下载压缩包文件到本地安装，我照他说的做了发现全部弄好了，而且又因为 python 包的问题，于是我又转向了 R。有点感慨读了三年生信，连安装个 R 包都安装不好，有点沮丧。在装好相应的 R 包之后，在实践代码过程也遇到了一些小的语法问题，这都是小问题，当然在 R 包使用的时候遇到不知道使用方法的情况，通过百度还是能解决的。然后就是结果分析的时候，当结果图出来的时候被惊到了，我选取的数据集也不大，怎么出现这么多疾病，在写论文的时候也是在想，我要补一下艾滋病的相关知识了，好好预防艾滋病。

在这里，要感谢自然语言处理课程的夏老师，他教会了我代码程序的另一种用法，以前写代码都是针对一些数字、字符串写编程，就算是处理文件也是数据格式文件，而非人类日常阅读的文章，自然语言处理课程教会了我更多地关于“自然语言”有趣的“玩法”，如词云，更快地掌握一篇文章的重点，还有一些在学术中能用到的技术，如 PubTator 获取 NCBI 的 PubMed 文章摘要，甚至获取全文。也要感谢实验课的师兄师姐，在我们 run 不出代码的时候帮我们解决问题，最后要感谢那些造轮子的大佬，是因为他们才有了更方便更快捷的相关软件和网站，不需要过高的技术要求，就可以得到相应的结果。

7 附录

S1. 课程项目实验代码和结果文件(github 链接)https://github.com/lunyiy/bioNLP/tree/main/Course-project

参考文献

[1] Heijden W, Wijer L, Keramati F, et al. Chronic HIV infection induces transcriptional and functional reprogramming of innate immune cells[J]. JCI insight, 6(7):145928, 2021.

[2] Fanales-Belasio E, Raimondo M, Suligoi B, et al. HIV virology and pathogenetic mechanisms of infection: a brief overview[J]. Annali Dellistituto Superiore Di Sanità, 46(1):5-14, 2010.

4.2 杨晓龙、周旺《HPO 富集分析》

帕金森病 (PD) 是威胁中老年人的“第三杀手”，并呈现明显的年轻化趋势，给家庭和社会带来了沉重的负担 [1]。大脑使用多巴胺来发送信息和协调控制运动，包括从走路到说话，写作甚至微笑都与其相关。大脑中负责多巴胺的细胞异常会导致帕金森病。我们目前并不清楚多巴胺细胞丢失的原因，但是研究人员们正在努力寻找保护这些细胞的方法。本研究使用 HPO 富集分析 [2]，从 NCBI 下载人类帕金森相关基因，并从分析结果中找到 HPO 注释与帕金森的一些联系，使我们更加了解这种疾病。

课程论文 GitHub 网址：`https://github.com/XiaolongYang-HZAU/HPO-Enrichment-of-NLP-Course`

HPO 富集分析

杨晓龙 [1], 周旺 [2]

[1] 华中农业大学信息学院，生信 1802，2018317220204
[2] 华中农业大学生命科学技术学院，2020304120210

摘要

帕金森病 (PD) 是威胁中老年人的“第三杀手”, 并呈现明显的年轻化趋势，给家庭和社会带来了沉重的负担 [1]。大脑使用多巴胺来发送信息和协调控制运动，包括从走路到说话，写作甚至微笑都与其相关。大脑中负责多巴胺的细胞异常会导致帕金森病。我们目前并不清楚多巴胺细胞丢失的原因，但是研究人员们正在努力寻找保护这些细胞的方法。本研究使用 HPO 富集分析 [2]，从 NCBI 下载人类帕金森相关基因，并从分析结果中找到 HPO 注释与帕金森的一些联系，使我们更加了解这种疾病。

关键词: 帕金森，HPO 富集分析，表型特征

1 课题概况

选修本课程的目的是根据以往上过本课的同学推荐，此课讲授的内容全部都是生信较为前沿的知识，贴合实际工作的内容，而且老师亲和，授课水平很高。

选择本课题大致有两个原因。第一是因为其难度较为适中，且 HPO 的注释集相比 GO 更加宽泛，尽管分析过程相比已经十分成熟的 GO 分析可能更加复杂，但是其结果可能更易于理解。第二是是因为家中有老人疑似得了此病，曾搜索过相关科普视频，了解到目前并没有根治帕金森的方法，因此想借着本次机会自己动手完成此次针对它的 HPO 分析，了解帕金森表型特征以及与 HPO 中其他表型异常间的关系，并获取相关的基因集表格，加深对此病的理解并且为进一步的分析做准备。

2 数据

首先在 NCBI Gene 数据库下载关于帕金森病的相关 symbol 资料，提取下载得到的 symbol 列，即为基因编号，保存为 unicode（utf-8）格式，并删去题头得到文件 symbol_Parkinson.txt，共有 1 列，345 行。

人类表型本体论 (HPO) 提供了在人类疾病中遇到的表型异常的标准化词汇。HPO 中的每个术语都描述了一种表型异常。本文的分析中用到了 HPO 的两类数据，分别是 hpo.obo 和 phenotype_to_genes.txt。hpo.obo 用于构建网络图数据文件，phenotype_to_genes.txt 用于匹配 HPOterms 的基础文件，两者都可从 HPO 官网下载。

3 研究方法

3.1 研究方法的算法背景，与其他方法的联系与区别

基因富集分析是分析基因表达信息的一种方法，富集是指将基因按照先验知识，也就是基因组注释信息进行分类。人类有约 30,000 个基因，总碱基对的数量约 32 亿。目前约有 3.2 亿可能的碱基对变异情况，

而每两个人之间的差异约为 2 千万个碱基对，也就是总碱基对的百分之 0.6。换句话说，人与人之间的基因序列相似度高达百分之 99 以上。这些细微的差别，导致了我们长得不同，性格也不同。那么怎么更好的理解这些不同呢？可以按照功能、通路等性质将基因做划分，这也是基因富集分析的重要作用之一。下面介绍一下 Fisher 精确分析多重检验校正的 BH 方法。

1）Fisher 精确分析 [3]

Fisher 精确检验是基于超几何分布计算的，它分为两种，分别是单边检验（等同于超几何检验）和双边检验，应用于将对象分成两组后的分类数据，以检查两组分类间是否有显著关系。

2）采用 BH 法 [4] 校正多重检验的 p 值：做 m 次无效假设作物的数量，如下表 1 所示：

表 1: Number of errors committed when testing m null hypotheses

	Decleared non-significant	Declared significant	Total
True null hypotheses	U	V	m_0
Non-true null hypotheses	T	S	$m-m_0$
	m-R	R	m

那么，被错误地拒绝了的无效假设的比例：

$$Q = V/(V+S) = V/R \tag{1}$$

FDR 值就是 Q 的期望值

$$E(Q) = E(V/R) \tag{2}$$

当同一个数据集有 n 次（n>=2）假设验证时，要做多重假设验证校正，BH 校正正是对每个 p-value 做校正，转换为 q-value。

$$q = p * n/rank \tag{3}$$

其中 rank 是指 p-value 从小到大排序后的次序。

3.2 研究方法中的核心思路

1、从 NCBI 上下载 symbol 数据，并自行构建 python 函数完成 symbol 编号与 ENSG 编号的转换，方便使用工具包。

2、调试并适当修改调整工具包的代码，使其能够完成本项目的 HPO 分析工作，获取分析结果表格。

3、根据分析结果绘制泡泡图与网络图。

4、对结果进行详尽注释的检索和进一步的生物学分析。

3.3 本文的方法部分与课堂讲授内容的联系和区别/或补充

课堂上并没有讲授 HPO 分析的具体代码实现，本文在 python 上实现了类似于课堂讲授的 GO 富集分析 [5] 的过程，且获得了类似的分析结果，实验设计部分更加复杂。

4 算法实践和代码编写要求

4.1 任务描述

根据已有的 python 包，进行适当修改，增添代码后完成从 NCBI 下载目标 Gene Symbol 并完成 HPO 分析的流程。最终得到分析结果的 csv 表格和泡泡图与网络图。

4.2 实验设计

1. 环境配置 WIN10 下的 VSCODE + Powershell 终端 +Python3.7.4 64bit。

2. 数据集划分和数据筛选与人类帕金森相关的基因的整个 symbo 列都作为输入数据，从 NCBI 下载目标 Gene Symbol 保存为.csv 文件，读取文件。按照 padj<=0.5 筛选出所需要的数据。

3. 格式转换将基因的 ENS 格式转化为 HPO 需要的 Entrez 格式。

4. 算法实施由于 fisher 检验和 BH 矫正方法都已经非常成熟，本文算法部分都采用了现成的 python 包，没有自行编写新算法。

5. 代码运行采用 R 语言和 Python 语言编写代码和运行代码。

6. 进行富集分析并将结果可视化运行执行富集分析的代码，并将结果可视化为泡泡图和网络图两种。

4.3 某些关键代码

本项目的 github 链接为：https://github.com/XiaolongYang-HZAU/HPO-Enrichment-of-NLP-Course

第一块：

```
代码来源：生信1802杨晓龙 原创
#将symbol编号转化为ENSG编号
result = []
  for i in out:                          #获取对应的ENSG编号列表
      if 'ensembl' in i.keys():
          genelist = i['ensembl']
          if type(genelist) == list:
              for j in genelist:
                  result.append(j['gene'])
          else :
              result.append(genelist['gene'])
```

第二块：

```
代码来源：生信1802杨晓龙 原创
#这个函数能生成main函数需要的前置文件
gsea = GSEA()
gsea.enrich(ENTRZ)
gsea.multiple_test_corretion(method='fdr_bh') #采用BH法矫正p值
print(gsea.enrichment_table.head(1))
gsea.enrichment_table.shape[0]
gsea.filter(by='padj', threshold=0.01)#筛选padj小于0.01
#print(type(gsea.enrichment_table))
t=gsea.enrichment_table
t.to_csv("enrichiment_table.csv")#保存分析结果表格为enrichiment_table.csv
```

第三块：

```
代码来源：https://github.com/Nanguage/BioTMCourse/tree/master/HPO20enrich
for term_id in tqdm(possi_terms):  # calculate the p-value of each possible terms
                genes = list(self.gaf[self.gaf.HPO_Term_ID == term_id].entrez_gene_symbol) #提取具有相同id的
                    所有term方便后续计算
                related_genes.append(genes)
```

```
            counts = self._get_counts(term_id)
            all_counts.append(counts)
            pval = self._calc_pvalue(*counts)
            pvals_uncorr.append(pval)
        study_count, n_study, population_count, n_population = zip(*all_counts)
```

5 主要的生物信息学实验和实验结论

5.1 实验结果

将筛选并进行格式转换后的数据进行 HPO 富集分析，结果保存为 enrichiment_table.csv, 具体为如下图 1。其中 HPO_term_ID 和 HPO_term_name 是 HPOterm 编号和注释标题，详细的注释信息附在草稿文件中；padj 是矫正后的 p 值，此值越小，则 term 与输入基因集关联程度越大。将筛选并进行格式转换后的数据进行 HPO 富集分析，富集分析结果筛选阈值是 padj 值小于 0.01，并对筛选结果按 padj 从小到大排序，结果保存为 enrichiment$_table.csv$, 1 HPO_term_ID HPO_term_name $HPOterm$ $github$ $padj$ p te

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1		HPO_term	HPO_term	gene_num	study_coun	n_study	population	n_populati	gene_ratio	backgroun	odd_ratio	pvalue	padj	related_genes													
2	2	HP:000130	Parkinsonis	155	74	10814	155	234119	0.006843	0.000662	10.33595	6.93E-56	1.65E-52	A2M MYORG ADH1C PCDH19 PLA2G6 ALS2 APOE APP APP PRKRA PRKRA ATP1A3 ATP1A3 SYNJ1 SQSTM1 CAT XPR1 LYST JAM2 CLN3 FBXO7													
3	105	HP:000206	Bradykines	162	75	10814	162	234119	0.006935	0.000692	10.02297	2.27E-55	2.70E-52	ACTA1 MYORG ADH1C PCDH19 PLA2G6 PLA2G6 PLA2G6 SLC25A4 PRKRA PRKRA PDE8B PDE8B ATP1A3 ATP1A3 SYNJ1 SYNJ1 SYNJ1 JAM2 FB													
4	118	HP:000206	Rigidity	203	74	10814	203	234119	0.006843	0.000867	7.89198	1.19E-45	9.43E-43	AARS1 ACTA1 ADAR ADH1C PLA2G6 PLA2G6 CASK PDE8B PDE8B ATP6V1A SYNGAP1 SYNJ1 SYNJ1 SYNJ1 SYNJ1 BRAT1 CACNA1A CACNA1B													
5	113	HP:000217	Postural ins	106	50	10814	106	234119	0.004624	0.000453	10.21209	6.07E-38	3.62E-35	ADH1C PLA2G6 PLA2G6 ABCC6 ATP1A2 ATP1A3 ATP1A3 RNASEH1 SYNJ1 SYNJ1 SYNJ1 CACNA1A CACNA1A CACNB4 ERCC8 ERCC8 CLN5 FB													
6	0	HP:000072	Dementia	222	68	10814	222	234119	0.006288	0.000948	6.631415	1.48E-36	7.08E-34	A2M ADH1C AARS2 ABCD1 APOE APOE APP APP APP APP APP APP GBA2 GBA2 ARSA ASAH1 ATP6V1A ATP6V1E1 CTSF ATP7B AUH SC													
7	111	HP:010031	Lewy bodi	31	28	10814	31	234119	0.002589	0.000132	19.5545	1.54E-34	6.10E-32	ADH1C PLA2G6 FBXO7 SNCAIP GIGYF2 C19ORF12 RAB39B EIF4G1 EIF4G1 GBA GBA GBA GLUD2 GRN MAPT NR4A2 VPS13C LRRK2 LRRK2 ATX													
8	102	HP:000232	Resting tre	54	33	10814	54	234119	0.003052	0.000231	13.23032	1.62E-30	5.51E-28	ADCY5 ADCY5 ADH1C SLC25A4 ATP1A3 SYNJ1 CACNA1G RRM2B SNCAIP GIGYF2 GIGYF2 DNAJC6 RAB39B DNMT1 ATP6AP2 ATP6AP2 EIF4G1													
9	107	HP:000071	Depressivit	483	88	10814	483	234119	0.008138	0.002063	3.944444	4.27E-28	1.27E-25	SMC1A ADH1C AARS2 PLA2G6 PLA2G6 KCNT1 KCNT1 ANG SLC25A4 PRSS12 ANXA11 AR ARSA ARVCF JRK ABCB11 ATP1A3 ATP1A3 STX16 C													
10	172	HP:000254	Parkinsonis	27	23	10814	27	234119	0.002127	0.000115	18.44227	2.76E-27	7.31E-25	PLA2G6 FBXO7 GIGYF2 GIGYF2 EIF4G1 GBA GCH1 PDE10A HTRA2 PARK7 MAPT MAPT MAPT POLG POLG LRRK2 LRRK2 VPS35 SNCA TAF1 TAF													
11	564	HP:000074	Apathy	140	45	10814	140	234119	0.004161	0.000598	6.958807	1.01E-25	2.41E-23	ACAT1 SMC1A SYNJ1 SQSTM1 SQSTM1 SQSTM1 FOXH1 FOXH1 FOXH1 CACNA1A ATXN10 CP CHMP2B CHMP2B GIGYF2 DNAJC6 DCT													
12	98	HP:000073	Hallucinatio	144	45	10814	144	234119	0.004161	0.000615	6.765507	3.85E-25	8.34E-23	ADH1C APP ARSA ARSA ARSA ARSA BCKDHA BCKDHB BCS1L BMPR1A SYNJ1 SQSTM1 CHI3L1 CLN5 WHRN COMT CPOX SNCAIP CTSH CYP27													
13	170	HP:000214	Frontotem	50	28	10814	50	234119	0.002589	0.000214	12.12379	1.28E-24	2.54E-22	PLA2G6 PLA2G6 SQSTM1 SQSTM1 SQSTM1 CCNF CHMP2B CHMP2B CHMP2B CYLD DCTN1 FUS GRN GRN GRN HNRNPA1 HNRNPA2B1 HNRN													
14	362	HP:000073	Disinhibitio	43	25	10814	43	234119	0.002312	0.000184	12.58699	1.08E-22	1.97E-20	APP SQSTM1 SQSTM1 SQSTM1 CHMP2B CHMP2B ABCA7 ABCA7 TOMM40 FMR1 FTL FUS GRN GRN HTT MAPT MAPT MAPT TBK1 TBK1 TREM													
15	115	HP:000075	Personality	78	32	10814	78	234119	0.002959	0.000333	8.881896	1.69E-22	2.88E-20	ADH1C PLA2G6 PLA2G6 APOE ATP7B SQSTM1 SQSTM1 EIF2B4 EIF2B3 EIF2B2 EIF2B5 MMACHC CHMP2B CHMP2B CHMP2B SNCAIP DCTN1 TIM													
16	289	HP:000071	Agitation	106	35	10814	106	234119	0.003237	0.000453	7.14846	8.73E-21	1.39E-18	ACAT1 ANG ANXA11 APP SYNJ1 SQSTM1 BRAT1 CFAP410 CACNA1A CCNF VAPB COX10 GABBR2 GABBR2 CHMP2B GIGYF2 MATR3 DAO DNA													
17	108	HP:001196	Substantia	16	15	10814	16	234119	0.001387	6.83E-05	20.29652	1.41E-19	2.10E-17	ADH1C FBXO7 SNCAIP GBA GLUD2 MAPT ATXN3 ATXN3 ATXN3 NR4A2 PRKN LRRK2 ATXN2 ATXN8OS SNCA TBP													
18	281	HP:000339	Muscle spa	171	42	10814	171	234119	0.003884	0.00073	5.317451	3.59E-19	5.03E-17	ACADM ACADVL AMPD1 AMPD3 ANG SLC25A4 ANXA11 AR AR ATP1A1 STX16 SYNJ1 SQSTM1 CFAP410 CACNA1S CRPPA CASQ1 CASQ1 CA													
19	589	HP:003022	Perseverat	34	20	10814	34	234119	0.001849	0.000145	12.73507	1.43E-18	1.89E-16	SQSTM1 SQSTM1 CHMP2B CHMP2B TACO1 NPC2 FUS GRN GRN GRN MAPT MAPT MAPT TBK1 NONO TREM2 TREM2 TMEM106B TMEM106B													
20	308	HP:000235	Memory in	150	38	10814	150	234119	0.003514	0.000641	5.484571	5.50E-18	6.90E-16	KLRC4 MYORG ABCD1 ABCD1 APOE APOE APP APP FAS ARSA BMPR1A SQSTM1 EIF2B4 EIF2B3 EIF2B2 EIF2B5 ACSF3 C4A XPR1 CCR1 CP MMA													
21	104	HP:000133	Dystonia	466	70	10814	466	234119	0.006473	0.00199	3.252089	6.44E-18	7.67E-16	ACADS DNAJC19 ACOX1 ADAR ADAR ADAR ADCY5 ADCY5 MYORG TBC1D24 TBC1D24 ADH1C AARS2 HACE1 HACE1 PLA2G6 PLA2G6 PLA2G													

图 1: 富集分析结果图。图片来源：部分实验结果在 Excel 中的截图

5.2 部分 HP

通过查询富集结果的编号，我们获得到了更详细的注释内容，我们对其展开分析，发现富集结果的注释主要包括以下两种：

1）性状注释直指帕金森，是帕金森人尽皆知的性状，例如：

Parkinsonism HP:0001300

由中脑黑质多巴胺生成细胞变性引起的特征性神经异常，临床表现为震颤、僵直、运动迟缓、行走和步态困难。

Parkinsonism with favorable response to dopaminergic medication HP:0002548

帕金森氏症是一种临床综合征，是许多不同疾病的特征，包括帕金森氏症本身，其他神经退行性疾病，如进行性核上麻痹，以及作为一些神经镇静药的副作用。一些但不是所有的帕金森病患者对多巴胺能药物有反应性，多巴胺能药物治疗后帕金森病的主要体征 (主要包括震颤、运动迟缓、僵硬和姿势不稳) 的改善显著减少。

Bradykinesia HP:0002067

运动迟缓 (Bradykinesia) 字面意思是运动缓慢，临床上用来表示运动执行缓慢 (与运动减退 (hypokinesia) 相对，后者用于表示运动启动缓慢)。

Substantia nigra gliosis HP:0011960

黑质胶质细胞局灶性增生

2）与帕金森可能存在密切联系，但也可能是其他病因导致的性状，例如：

Frontotemporal dementia HP:0002145

一种与额颞叶退化相关的痴呆，临床上与性格和行为变化如去抑制、冷漠和缺乏洞察力相关。额颞叶痴呆的显著特征是出现局灶性综合征，如进行性语言功能障碍、失语症或额叶功能障碍的行为改变。

Disinhibition HP:0000734

缺乏约束表现在几个方面，包括无视社会习俗、冲动和糟糕的风险评估。去抑制影响运动、本能、情绪、认知和知觉方面，其体征和症状类似于躁狂的诊断标准。性欲亢进，暴饮暴食，以及攻击性的爆发都是不受抑制的本能冲动的表现。

Depressivity HP:0000716

经常感到沮丧、痛苦和/或绝望; 难以从这种情绪中恢复过来; 对未来的悲观; 普遍的耻辱; 自卑的自我价值感; 自杀的想法和自杀行为。

5.3 实验结果可视化

1）可视化为泡泡图，下图 2 展示了前 20 个的可视化结果（10+10）。

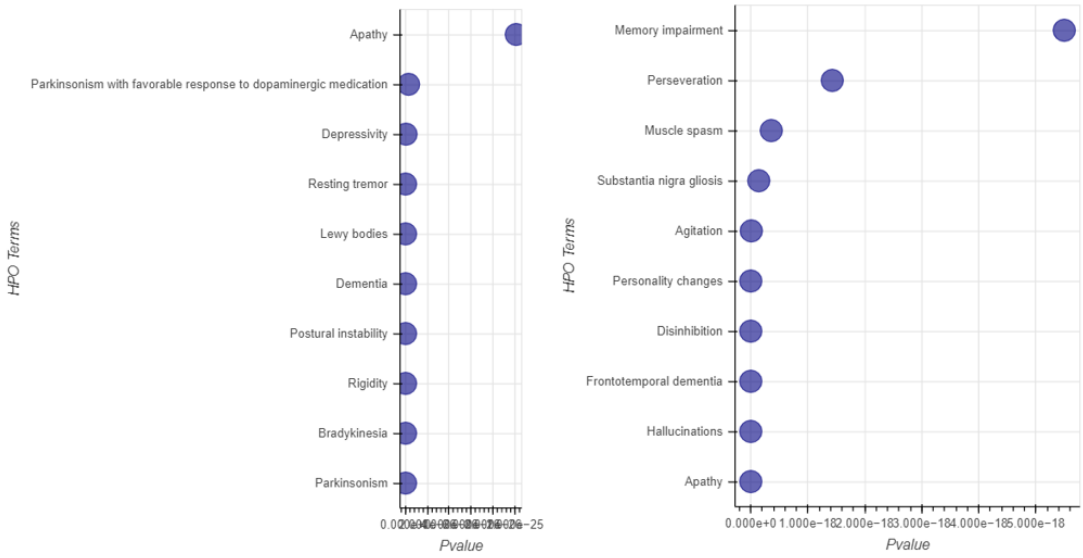


图 2: 前 20 个结果可视化泡泡图。图片来源：Python3.7.3 绘制

2）可视化为网络图，下图 3 展示了前十五个点的示意图。

5.4 实验结果分析

我们认为出现第一种类型的注释结果是因为 HPO 对帕金森的研究已经较多，它们的 padj 值也非常小，这类 term 对我们的价值在于，可以通过他们的 terms 基因集获取更多已知的与帕金森相关的基因。第二种类型的 terms 则可能具有更大的研究价值，我们可以通过其他手段从这类 terms 的基因集中获得以往不知道的与帕金森相关的重要基因。也可以收集其注释信息并进行语义分析，获取与帕金森相关的重要词汇以及词汇的上下文联系等等。

从泡泡图中可以看出 HPO 基因集中与输入基因集关联最大的 HPOterm 就是帕金森，这证明了输入数据的质量还不错，进一步保证了富集结果的准确可靠；其次，还富集到了许多相关表型的 terms，这有助于我们进一步理解帕金森的表型特征。从网络图中可以看出 padj 值最小的 15 个 HPOterms 与其他 terms 之间的联系，这有助于我们更详细地了解相关联的表型特征并进行扩展分析。

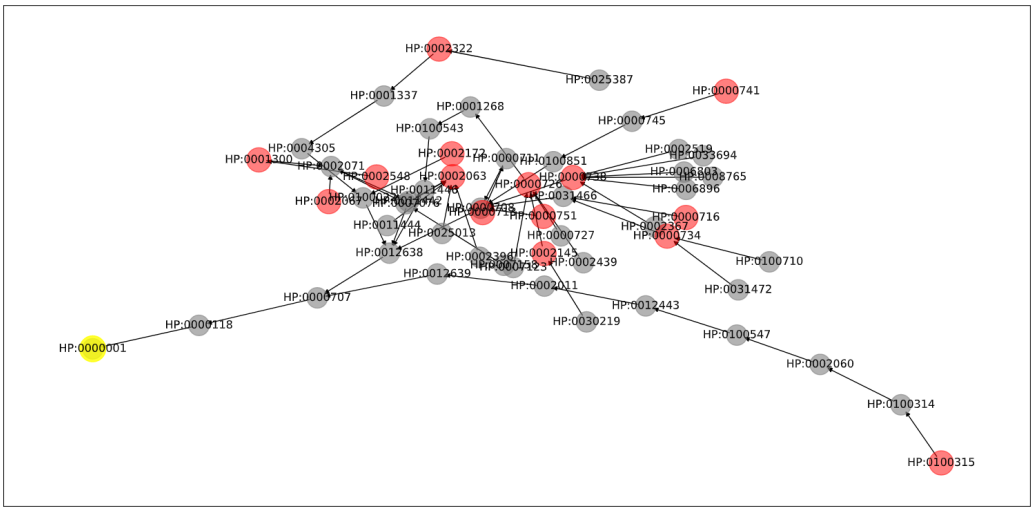


图 3: 前 15 个结果可视化网络图。图片来源：Python3.7.3 绘制

尽管通过 HPO 获得的表型结果分析是比较初步的，但是结果表格中不仅存在表型特征，还存在着各个 terms 中的关联基因，而这些基因可能会为对帕金森病的进一步研究做出贡献，如作为 GO 富集的输入基因集，进行进一步的关于分子层面的分析等等。

6 后记

6.1 课程论文构思和撰写过程

5 月中旬老师在微信群里发布 HPO 的辅助文件时本项目已经定下题目并且完成了基本的代码流程实现，因此我们优先选择了继续按照已有思路进行论文攥写。我们的思路是先收集 HPO 富集分析的相关论文和辅助包，获取相关文献和帮助并且对比各个方法的可行性和难易度，最终决定了本次的分析方法。尽管目前我们也查阅到了更好的，依赖于 R 包的实验方法，但是其源代码过于复杂，且 R 的代码编写我们并不如 Python 熟悉，综合考虑后我们还是决定使用本文这种我们易于掌握的方法。在实际实施中，遇到了一些代码问题和环境配置问题。得益于算法比较简单易于理解，代码问题在查阅源代码并进行增删修改后都得到了妥善解决。而环境问题主要是因为我一开始想用 ubuntu20.0.4wsl（一个 win10 下的 ubuntu 子系统）进行代码编译导致的，这个环境配置与一般的 linux 不太一样，而我又没有性能足够的 linux 服务器，所以花费了较多时间。在本论文中最终还是选择了 win10 环境编译执行代码。

我们先使用 markdown 编辑器记录了全部的实验操作流程（详见 github 项目），并在此基础上使用 markdown 构成论文的草稿，最后再将文本倒入 tex 编辑器的模板文件中，完成了此次论文的攥写。

6.2 所参考主要资源

1. 往届学长作业：https://github.com/tongliu-liu/HPO-enrichment-analysis

2.R 包:HPOSim:https://pubmed.ncbi.nlm.nih.gov/25664462/: : text=The%20Human%20Phenotype%20Ontology%20%28HPO%29%20provides%20a%20standardized,used%20offline%20and%20provide%20only%20few%20similarity%20measures

3.Python HPO 基因集合富集分析示例:https://nanguage.github.io/examples/hpo_enrich/example_sagd_00055.html

4. HPO 官网，下载 HPO 数据，查 HPO 编号和注释用：https://hpo.jax.org/app/

6.3 生物信息学实验设计的构思和体会

我们在构思本实验时，希望的到许多与帕金森相关的表型特征和可能与帕金森相关的基因集。事实上我们也得到了这种结果。但是如果要进行更严谨的富集分析，我们应该获取基因芯片数据，并进行基因差异分析后得到上调下调基因，分别进行富集分析。然而更具以往项目的经验，在个人电脑上从上游芯片数据开始分析会耗费大量计算时间，且对帕金森的基因研究已经有较多的结果，我们从上游开始分析结果也不一定会更好，因此我们选择了直接从 NCBI 下载基因集。这次分析流程让我加深了对 HPO 富集的认识，这会是宝贵的经验。但是受限于篇幅，题目和时间，我们并没有做后续的分析，如对注释的语义分析，对富集出的 terms 基因集的分析等等。我们可能会在结课后继续沿着这两个方向进行研究，继续深入进行下游的分析流程。

6.4 人员分工

实验设计，数据收集，代码设计，论文起草：生信 1802 杨晓龙
早期资料收集，正式论文攥写与排版：周旺

7 附录

S1. 2021 年课程网页. `https://hzaubionlp.com/course-bionlp-and-kd/`

S2. 课程论文可以使用的基础代码，可参考 GitHub 页面, `https://github.com/XiaolongYang-HZAU/HPO-Enrichment-of-NLP-Course`

参考文献

[1] 焦倩 and 姜宏. 帕金森病病因与发病机制研究现状及其诊治意义. *青岛大学学报 (医学版)*, 57(02):159–162, 2021. 页数: 4.

[2] 孙艳. *毛细管电泳富集技术在食品中几类药物残留分析中的应用*. PhD thesis, 浙江工商大学, 2015. 期: 05.

[3] 刘法贵 and 李亦芳. n 维 fisher 方程的精确解及定性分析. *吉首大学学报 (自然科学版)*, (06):4–6+8, 2007. 页数: 4.

[4] 张凤松. bh 法观测采区煤巷底板坡度. *淮南职业技术学院学报*, (03):23–25, 2005. 页数: 3.

[5] 葛杰, 贾月辉, 李继媛, 王琪, 韩云峰, 谢志平, 杨晓蕾, and 郑毅. 基于富集分析的乳腺癌发病相关通路探讨. *系统医学*, 5(11):29–33, 2020. 页数: 5.

4.3 杨迟、屈伸洋《HPO 富集分析》

精神分裂症 (Schizophrenia，SCZ) 是一种原因未明的慢性疾病，临床上往往表现为症状各异的综合征，具有遗传和神经生物学的异质背景。发病症状多表现为认识功能障碍、幻觉 (精神病症状) 等。目前鉴定与 SCZ 有关的因果基因仍具有很大的挑战。本次实验基于 HPO 富集分析，采用文献中的 10 组数据进行了差异表达分析和富集分析。通过设计 HPO 富集算法以及搜集临床实验资料，推测可能与 SCZ 有关的基因和致病机制。

课程论文 GitHub 网址：`https://github.com/PeachYang123/HPO-Enrichment`

HPO 富集分析

杨迟 [1], 屈伸洋 [2]

[1] 华中农业大学信息学院，430070，武汉，湖北，中国
[2] 华中农业大学信息学院，430070，武汉，湖北，中国

摘要

精神分裂症（Schizophrenia，SCZ）是一种原因未明的慢性疾病，临床上往往表现为症状各异的综合征，具有遗传和神经生物学的异质背景。发病症状多表现为认识功能障碍、幻觉（精神病症状）等。目前鉴定与 SCZ 有关的因果基因仍具有很大的挑战。本次实验基于 HPO 富集分析，采用文献中的 10 组数据进行了差异表达分析和富集分析。通过设计 HPO 富集算法以及搜集临床实验资料，推测可能与 SCZ 有关的基因和致病机制。

关键词: SCZ,HPO, 疾病, 富集分析, 差异表达分析

1 课题概况

HPO 全称 Human phenotype ontology 是旨在提供人类疾病中用于描述表型异常的标准词汇的人类表型术语集，目前已经成为了表型交换的全球标准。精神分裂症是一种常见的精神疾病，成因复杂并且与许多疾病相关联，目前对于 SCZ 相关基因和关联疾病的研究还在进行中。本课题基于超几何分布和 FDR 校验计算原理，设计算法以实现差异表达基因在 HPO 术语上的富集，来挖掘 SCZ 差异基因和疾病表型之间的联系，并与已知的症状表型相比对以验证富集的准确性，同时寻找与 SCZ 有关联的潜在分子信息。

2 数据

本实验采用的数据来自 Molecular signaling pathways underlying schizophrenia [1]，研究人员重新编辑诱导多能干细胞（iPSC），将 iPSC 区分为皮质神经元和易感性兴奋神经元，对精神分裂症患者的神经元转录组变化进行 RNA 测序。

测序平台：Illumina NextSeq 500；

测序对象：iPS 衍生神经元；

原始 RNA-seq 数据来自：GSE174704；包含了 5 个健康人类神经元细胞转录组数据作为对照组，5 个患精神分裂症人类神经元细胞转录组作为实验组；

参考基因组：R 中的基因注释包 org.Hs.eg.db；

基因表达矩阵：GSE174704_gene_counts.txt.gz；

HPO 背景数据集：下载 HPO 官网最新数据

`http://purl.obolibrary.org/obo/hp/hpoa/genes_to_phenotype.txt`。

3 研究方法

3.1 研究方法的算法背景，与其他方法的联系与区别

（1）富集分析

基因功能富集分析手段常用于揭示基因的生物分子机制，其算法基本思想来源于 over-representation analysis （ORA），包括了超几何分布、卡方检验、二项式分布等统计检验方法。除了 ORA，常用的富集分析方法还有 functional class scoring（FCS）、pathway topology（PT）、network topology（NT），在此只做概括性描述。

本实验主要采用的富集方法：超几何分布。

$$P_Q = \frac{C_M^Q * C_{N-M}^{K-Q}}{C_N^K} \tag{1}$$

N：HPO 背景数据集中所有 gene 数目; M：HPO 背景数据集中某个 Term 的基因数目；K：差异表达基因数目；Q：该 Term 在差异基因中对应的基因数目；

$$p = 1 - \sum_{i=0}^{Q-1} P(i) = P(Q) + P(Q+1) + ... + P(K) \tag{2}$$

对于给定一个 HPO Term t 和一组基因集合, 假设有 K 个基因注释在整个 HPO 中，并且有 Q 个是被 Term t 注释的基因。因此，可以通过计算每个 Term 的累计概率计算 P-Value，根据 P-Value 由小到大筛选所富集的 Term，P-Value 越小则富集效果越显著。

（2）FDR 校正

假设检验的基本思路时设立零假设 null hypothesis（H0）和 alternative hypothesis（H1），首先设置 H0 假设成立为前提条件，计算 H0 发生的概率，所这个概率很低，我们就认为小概率事件为不可能发生事件，拒绝 H0，接受 H1。但是不排除 H0 在很小的概率下也发生，而我们误以为 H1 发生，此时就需要多次统计推断引入 Q-Value 衡量假阳性率。FDR 校正的目的就是控制 Q-Value 通过多重检验校正减少假阳性发生次数，最常用的方法就是 Benjaminiand Hochberg（BH）法。

3.2 研究方法中的核心思路

本文旨在实现差异表达基因在 HPO 数据集上的富集，挖掘基因和疾病表型之间的联系，再与已知的表型比较可以得出富集的准确性或者预测一些潜在的疾病表型。

3.3 本文的方法部分与课堂讲授内容的联系和区别/或补充

本文中实验主要采用 ORA 方法，即传统的富集分析流程：获取差异表达基因、富集分析和结果解读。我们设计了在使用超几何分布计算 P-Value 后用 BH 法对结果进行多重假设检验以降低假阳性概率，最后对富集结果进行可视化与注释。

为了寻求改进我们探究了其他几种基因功能富集分析方法。ORA 法的大致内容是使用基因数目信息设置阈值进行筛选得到差异表达基因，计算得到的基因列表与通路中的基因集重叠基因数目，以统计检验的方式检验这个数目是否显著高于随机，结果稳健可靠。缺点在于，该方法仅关注到基因的数目信息，需要设置阈值人为筛选差异基因，因此阈值的选择一定程度上左右了关键基因是否被漏选使其灵敏性下降，同时该方法也未能关注到基因间复杂的相互作用关系。FCS 在 ORA 的基础上做了改良，输入值为根据基因表达水平差异值打分排序后的基因表达谱，并且通过统计模型它能够将待测基因集分数转化为功能集分数，纳入基因表达值属性。PT 法则弥补前两者缺陷，考虑通路中基因位置属性，例如上下游基因改变对通路影响明显不同等。GO 等注释数据库中基因功能集中不包含任何拓扑结构信息，仅提供了可能属于同一通路的所有基因列表。所以，PT 方法不能被用于 GO 通路的富集分析。NT 法顾名思义，是一种基于生物网络拓扑结构的富集分析方法，与传统方法比较，NT 法考虑了系统层面基因重要性程度和关联信息，大大提高了预测结果准确度。

4 算法实践和代码编写要求

4.1 任务描述

根据上述研究方法中所述的富集主要采用超几何检验的统计学方法，通过公式 *(1)*，需要先计算出对应的 N,K,M,Q 值，再计算每个 Term 的 PValue，经过 FDR 矫正后筛选富集条目。

4.2 实验设计

（1）软硬件条件

platform:x86_64-w64-mingw32;

version:R version 4.0.5;

（2）数据集划分

使用 R 设计代码提取差异表达的基因，主要使用到的 R 包有 DESeq2。以数据框形式载入基因表达表，condition 分为健康和患病两组以此划分数据集，计算 log2FoldChange、p-value 和矫正后获得的 padj 值，设置筛选条件 padj 小于 0.1、log2FoldChange 大于 0.6 即可获得 46 个差异表达基因。

（3）数据预处理

得到差异表达基因后，将其 SYMBOL 转换为 ENTREZID，并且只保留 ENTREZID 列，在 EXCEL 中去重后仅剩 42 个基因，保存为 df_gene.csv。

在 HPO 官网所下载的 genes_t_phenotype.txt 文件有 9 列，分别为 entrez-gene-id，entrez-gene-symbol，HPO-Term-ID，HPO-Term-Name，Frequency-Raw，Frequency-HPO，Additional Info，from G-D source，G-D source，disease-ID for link，只保留需要的 3 列 entrez-gene-id，HPO-Term-ID，HPO-Term-Name，在 EXCEL 中去重后保存为 genes_to_phenotype_edit.txt。

（4）算法实施

由于 HPO 背景数据集中 entrez-gene-id 和 HPO-Term-ID 是多对多的关系，需要先实现 ENTREZID to HPOID、HPOID TO ENTREZID 的一对多关系提取，才能便于之后的富集计算。

N 值是 HPO 背景数据集的 ENTREZID 去重计算后的数值；K 值是差异基因与 HPO 背景数据集基因的交集，经过计算，42 个差异基因中仅有 18 个在 HPO 中；M 值是一个记录每个 Term 在 HPO 所对应的基因数目的向量，通过循环访问 HPOID TO ENTREZID 计数得到该向量；Q 值是一个记录每个 Term 在差异基因中相对应数量的向量，通过将差异基因的所有匹配的 HPOID 列成列表，再用循环计数每个 Term 出现了多少次该 Term 的 HPOID 可以得到 Q 的值。

计算完所需的四个参数的数值，代入 P-Value 计算式，调用 fdrtool 包矫正后，筛选 Q-Value 较小的条目，用 ggplot 绘制柱状图。

4.3 某些关键代码

```
代码来源：http://xiajingbo.weebly.com/uploads/1/3/3/0/13306375/enrich.pdf
#p值计算
#参数解释，在某个HPO term中：N:HPO中的所有gene数目；K:导入的所有差异基因数目；M:该term在HPO中对应的基因数目；Q：该term在差异基因中对应的基因数目
GetPValue <- function(Q,M,N,K)
  1 - sum(sapply(0:(Q - 1), function(i) choose(M, i) * choose(N - M, K - i)/ choose(N, K)))
```

```
代码来源：原创代码，已上传GitHub
https://github.com/PeachYang123/HPO-Enrichment
#计算N,K值
N <- length(HPO_GENE_UN$HPO_Data_Set.entrez.gene.id)
K <- length(Gene_Set$ENTREZID)
```

```
6
7  #计算M
8  vM <- vector()
9  for (i in 1:length(HPOID_UN$HPO_Data_Set.HPO.Term.ID)){
10   vM[i] <- length(HPOID_to_GENE[[i]])
11 }
12
13 #计算Q
14 vQ <- vector()
15 #计算每个差异基因对应的所有HPOID
16 Gene_Set_HPOID <- list()
17 num <- 1
18 for (i in 1:length(GENE_to_HPOID)){
19   for (j in 1:length(Gene_Set$ENTREZID)){
20     if(names(GENE_to_HPOID)[i]==as.character(Gene_Set$ENTREZID[j])) {
21       Gene_Set_HPOID[num] <- GENE_to_HPOID[i]
22       names(Gene_Set_HPOID)[num] <- Gene_Set$ENTREZID[j]
23       num <- num+1
24       }
25   }
26 }
27 num2 <- 0
28 Gene_Set_HPOID_Unlist <- unlist(Gene_Set_HPOID)
29 for (i in 1:length(HPOID_UN$HPO_Data_Set.HPO.Term.ID)){
30   for(j in 1:length(Gene_Set_HPOID_Unlist)){
31     if(HPOID_UN$HPO_Data_Set.HPO.Term.ID[i]==Gene_Set_HPOID_Unlist[j]){
32       num2 <- num2+1
33     }
34   }
35   vQ[i] <- num2
36   num2 <- 0
37 }
```

5 主要的生物信息学实验和实验结论

本实验主要包含的生物信息学实验有三个：差异表达分析、HPO 富集分析和后续生物学意义挖掘。

5.1 实验结果

我们总共富集到 42 个 HPO 图 *1*, 包括单侧肾发育不全（HP:0008718）、伤口愈合不良（HP:0001058）、脑动脉扩张（HP:0004944）等，可以看出我们富集出的表型与神经性疾病表型重叠程度较低，出现这种问题的原因我们推测可能是：目前发现的精神分裂症相关基因较少 [2]，提取差异基因时设置的阈值过高，导致关键基因被过滤；对于综合性疾病，其相关表型众多，是由多对微效基因协同与环境作用导致的复杂疾病 [3]，目前研究已发现患有精神分裂症的人群还可能患有免疫性疾病出现炎症等，因此我们富集的基因可能是通过复杂的生物代谢网络与 SCZ 关联。

在 string 网页上输入基因可以得到基因表达蛋白之间相互作用关系，根据网页分析结果用 Cytoscape 画出的图像图 *2*，发现关联强度较高的蛋白有：BGN、THBS1、COL4A1、TGFBI、TYRP1 等。BGN 为重组人类双糖链蛋白多糖，可能参与胶原纤维的组装，富含亮氨酸的重复蛋白聚糖；THBS1 是血小板反应蛋白，介导细胞与细胞和细胞与基质相互作用的粘附糖蛋白，在内质网应激反应中发挥作用，通过与激活转录因子 6（ATF6）的相互作用产生适应性内质网应激反应因子；IGFBP3 是胰岛素样生长因子结合蛋白，IGF 结合蛋白可延长 IGF 半衰期，能改变 IGF 与其细胞表面受体的相互作用等；TYRP1 为 5,6-二羟基吲

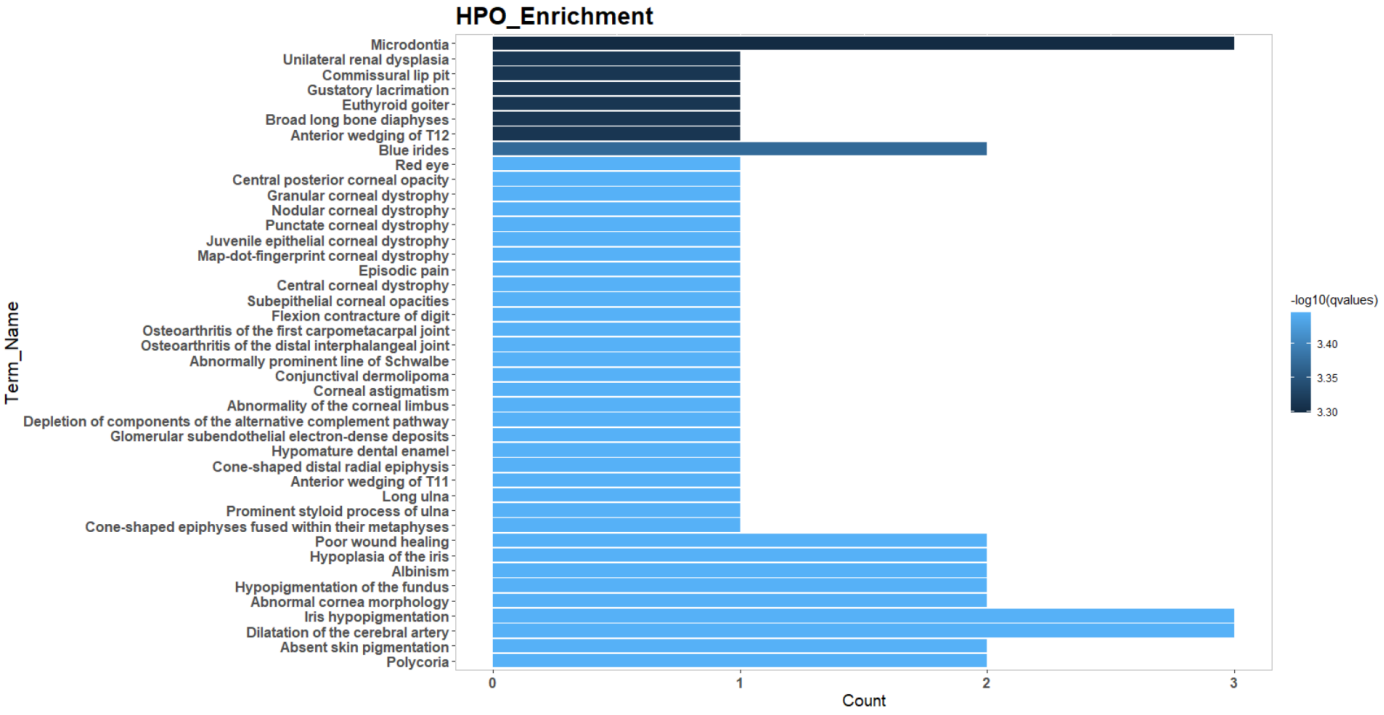


图 1: HPO 富集分析结果柱状图展示。图片来源：自绘；工具：R

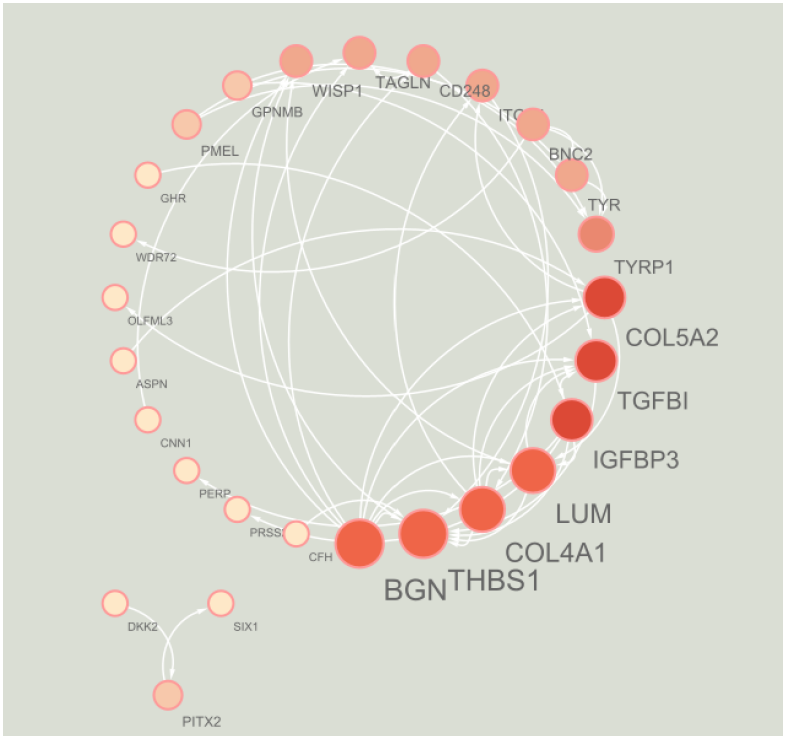


图 2: 基因表达蛋白互作图。图片来源：自绘；工具：Cytoscape

哚-2-羧酸氧化酶，能与其他因子结合调节影响黑色素的生成。查阅临床资料发现，神经黑色素也就是在中脑多巴胺神经元内产生的暗色素可能标志精神分裂患者多巴胺功能，这与差异表达的蛋白功能相关；血小板的异常表达可能影响着免疫系统的功能，免疫细胞黏附水平改变也与 SCZ 记忆障碍有关 [4]。

5.2 后续分析

由于 HPO 侧重实验组对照组基因表达差异，容易遗漏某些差异表达不显著但是有重要生物学意义的基因，因此我们对实验做出改进，采用 GSEA 重新考虑了基因生物特性和调控网络等信息，并绘制 KEGG 通路图，探究分子机制。最终得到的相关 KEGG 通路有：人类 T 细胞白血病病毒感染（hsa05166）、癌症转录失调、类风湿关节炎、心肌收缩、白细胞跨内皮迁移（hsa04670）、心肌细胞肾上腺素能信号（hsa04261）、流体剪切应力和动脉粥样硬化（hsa05418）等，除了嗅觉转导（hsa04740）受到激活，其他通路均为抑制。观察发现这些相关通路与 SCZ 临床症状相关性更强，研究其细胞中的调控通路可以为后续实验提供思路。

我们选择 hsa05418 绘制通路图图 *3*作为示例。

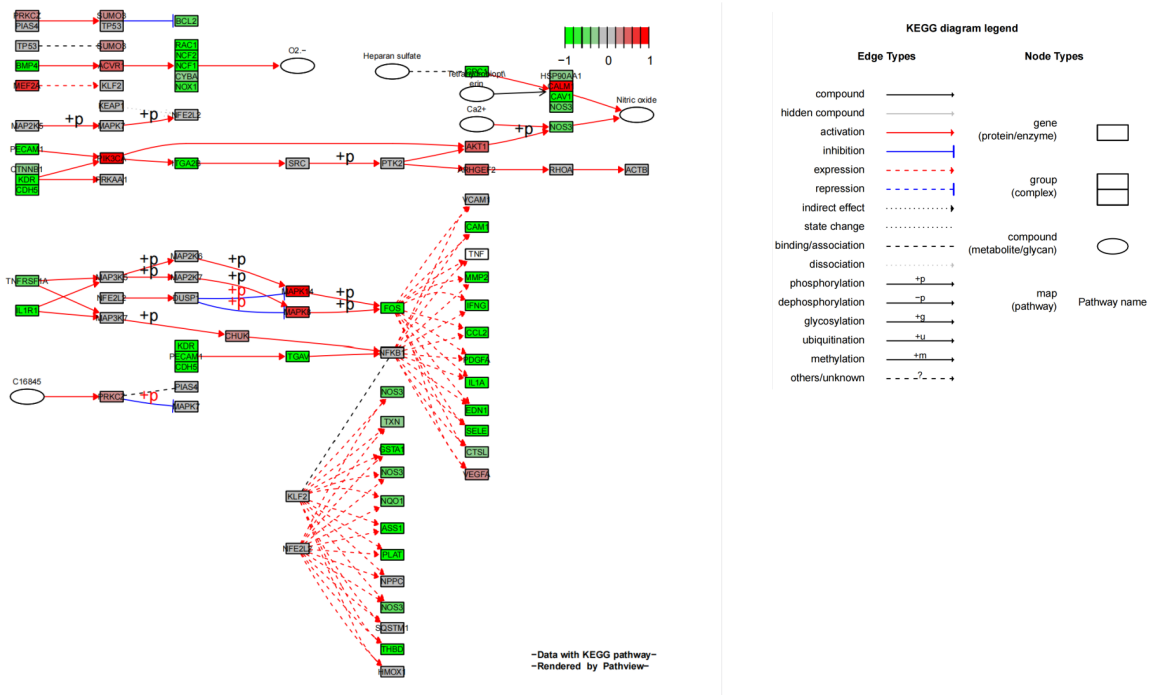


图 3: KEGG 通路图展示（hsa05418）。图片来源：自绘；工具：R

6 后记

6.1 课程论文构思和撰写过程

我们使用 Overleaf LaTex 的在线排版器，按照老师提供的模板分工撰写论文。在构思时，我们决定把重点放在代码原理的说明和实现上。在撰写过程中，由于是初次使用 LaTex 进行编辑，对于插入图片、公式等操作流程，没有以往在 word 中书写娴熟，但是 LaTex 严整有序的排版使得文章更具有美观性和规整性。

6.2 所参考主要资源

(1) 差异基因算法参考：https://www.jianshu.com/p/77846e847428
https://www.jianshu.com/p/b55276e46f0c
(2)差异基因代码参考:http://www.bioconductor.org/packages/release/bioc/html/DESeq2.html
https://zhuanlan.zhihu.com/p/30350531

(3) HPO 富集算法参考：https://pubmed.ncbi.nlm.nih.gov/25664462/
https://www.zhihu.com/question/36989716/answer/1158993121
(4) HPO 富集代码参考：http://xiajingbo.weebly.com/uploads/1/3/3/0/13306375/enrich.pdf
https://github.com/tongliu-liu/HPO-enrichment-analysis

6.3 代码撰写的构思和体会

在撰写提取差异基因代码时，我们首先明确原始数据提供的信息和缺失信息，设计数据预处理的步骤。总共 35431 个基因中只有 11 个基因通过筛选，这是因为筛选设置的差异表达倍数阈值通常为 2，这使得低差异表达量的关键基因可能被误筛，导致结果基因数量太少。因为这个问题我们才着手研究其他分析方式，学习全面考虑基因属性的富集计算方法。

在撰写 HPO 富集代码时，我首先参考了老师所提供的刘师兄的代码，在仔细研读之后，发现其中有许多参数的计算并没有在论文里说明清楚其含义。也因为我自身对于超几何检验算法理解不够深刻，我又查找了相关资料，设法去将 HPO 富集和经典模型中抓取的黑白球相互对应，在阅读了 HPOSim: an R package for phenotypic similarity measure and enrichment analysis based on the human phenotype ontology 这篇文献里对于富集分析算法的描述，才算是豁然开朗，明白了每个参数的意义。后来也偶然查找到了老师曾经写的代码记录，我将代码跑通了一遍后，研究了核心思路加上对算法的理解，我开始自己撰写代码。P-Value 的计算方程参考了老师的代码 GetPValue，参数的计算中由于列表名不能直接通过字符型变量匹配去提取想要的内容，只能通过暴力循环套循环实现计算过程，在实际的运行中，明显可以感觉我的代码运行时间较长，还有很多改进和优化的空间。

我曾尝试过调用上述文章中的 HPOSim 包，进行 HPO 富集，并与我们自己撰写的代码结果做对比试验，但是不幸的是，该包在四年前已经停止更新，下载最新的版本也无法载入到 4.0.5 的 R 当中，此外该包在网络上也搜索不到引用的资料和测试数据的分析，最终只能放弃这个思路。

6.4 生物信息学实验设计的构思和体会

本次实验的本质是应用课程所学算法，用这个算法能解决什么问题才是本次实验设计思路来源。在这种思路下，我们查阅了许多富集分析应用资料，借此了解到前沿分析方法。

在后期论文修改阶段我们改良了实验方法，降低阈值获得更多差异基因。对于神经性病症这种综合症而言，为了挖掘基因的分子机制还需要考虑基因在代谢通路、网络中的作用以及基因的空间位置结构，结合目前火热的 GSEA 和 NT 法，实验得到的结果与目前临床实验结果更加吻合。

6.5 人员分工

屈伸洋：SCZ 相关数据和背景资料的收集，差异基因的处理，HPO 富集结果分析，论文撰写；
杨迟：HPO 富集算法实现和代码撰写，论文撰写；

7 附录

S1. GitHub 链接. https://github.com/PeachYang123/HPO-Enrichment

参考文献

[1] Jari Tiihonen, Marja Koskuvi, Markku Lähteenvuo, Kalevi Trontti, Ilkka Ojansuu, Olli Vaurio, Tyrone D. Cannon, Jouko Lönnqvist, Sebastian Therman, Jaana Suvisaari, Lesley Cheng, Antti Tanskanen, Heidi Taipale, Šárka Lehtonen, and Jari Koistinaho. Molecular signaling pathways underlying schizophrenia. *Schizophrenia Research*, 232:33–41, 2021.

[2] Chaodong Ding, Chunling Zhang, Richard Kopp, Liz Kuney, Qingtuan Meng, Le Wang, Yan Xia, Yi Jiang, Rujia Dai, Shishi Min, Wei-Dong Yao, Ma-Li Wong, Hongyu Ruan, Chunyu Liu, and Chao Chen. Transcription factor pou3f2 regulates trim8 expression contributing to cellular functions implicated in schizophrenia. *Molecular psychiatry*, September 2020.

[3] Ana Costas-Carrera, Clemente Garcia-Rizo, Byron Bitanihirwe, and Rafael Penadés. Obstetric complications and brain imaging in schizophrenia: A systematic review. *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging*, 5(12):1077–1084, 2020.

[4] Helen Q. Cai, Thomas W. Weickert, Vibeke S. Catts, Ryan Balzan, Cherrie Galletly, Dennis Liu, Maryanne O'Donnell, and Cynthia Shannon Weickert. Altered levels of immune cell adhesion molecules are associated with memory impairment in schizophrenia and healthy controls. *Brain, Behavior, and Immunity*, 89:200–208, 2020.

5

基于水稻性状本体的水稻基因-性状关联挖掘

GO 富集之所以能成为普遍使用的方法，是因为标准化注释信息的可获取性。那么，如果换做一个从头开始的领域呢？

– Jingbo Xia

项目要求：

利用提供的水稻性状本体，对 PubMed 数据库提供的水稻文献进行挖掘，主要关注水稻基因-水稻性状之间的共句显示。

提示：

水稻性状本体数据：`https://github.com/bionlp-hzau/TOMapping`

使用 PubTator 获取基因实体。`https://www.ncbi.nlm.nih.gov/CBBresearch/Lu/Demo/tmTools/curl.html`

相关论文三篇：

孙梓淳祝宏涛《水稻基因-性状的文本挖掘》

赵柯韦、黄奇楠《水稻基因与性状的共句显示》

陶芳婷、黄婷婷《水稻基因-性状的挖掘》

5.1 孙梓淳祝宏涛《水稻基因-性状的文本挖掘》

如今，对于植物的研究文献在世界范围内迅速增长，通过对这些丰富的文本资源进行数据挖掘，我们可以更好地理解植物基因与性状之间的关联，从而给予一定的启发，并指导进行后续分析。在本项目中，我们使用 PubTator 对 18299 篇水稻文献的摘要中的实体进行挖掘，提取其中的 Gene 信息，并结合植物性状本体数据库进行匹配，从而找到基因与性状之间的关联，并做后续的研究。

课程论文 GitHub 网址：`https://github.com/passion-web/NLP`

水稻基因-性状的文本挖掘

孙梓淳 [1], 祝宏涛 [2]

[1] 华中农业大学信息学院生信 1802 班，430070，武汉，湖北，中国
[2] 华中农业大学信息学院生信 1801 班，430070，武汉，湖北，中国

摘要

如今，对于植物的研究文献在世界范围内迅速增长，通过对这些丰富的文本资源进行数据挖掘，我们可以更好地理解植物基因与性状之间的关联，从而给予一定的启发，并指导进行后续分析。在本项目中，我们使用 PubTator 对 18299 篇水稻文献的摘要中的实体进行挖掘，提取其中的 Gene 信息，并结合植物性状本体数据库进行匹配，从而找到基因与性状之间的关联，并做后续的研究。

关键词：PubTator、水稻、共现挖掘

1 课题概况

感谢刘同学的推荐，让我选到了这门课，初步了解到了自然语言处理的魅力所在。当前，尽管自然语言处理工作飞速发展，但对于生物文本挖掘的工作很少有进展。本文将对水稻文献进行挖掘，获取其中的 Gene 信息，并做粗略的统计学分析，尽可能的获取水稻研究中的热点基因，希望能够给相关领域的研究者提供一定的帮助。同时结合 PTO 数据库，在文本数据中找到基因与性状之间的关联，并结合网络图进行研究。

2 数据

数据 1. result_OZ_Pubtator.txt：PMID 号。我们最初是计划直接从 NCBI 上面下载，但发现最多只能下载一万篇。后面在夏老师的建议下，我们采用了 edirect 包中的 esearch 命令，获取文献所有的 pmid 号，总计 18299 篇。

数据 2. result_OZ_Pubtator.txt：我们在 Linux 系统下利用 PubTator 工具获取的结果文件。结果包括文献的 title、abstract、以及实体信息。

数据 3. TO terms：感谢夏老师提供的在 github 上提供的 1554 个 TO terms。https://github.com/bionlp-hzau/TOMapping。其内容包括 TO Term 的 id、name、synonym。

数据 4. result_fen.txt：PTO mapping 之后的结果文件。

数据 5. match.txt：基因与 PTO 共现挖掘之后的结果文件。

3 研究方法

3.1 研究方法的算法背景，与其他方法的联系与区别

PubTator 是一个在线的辅助生物精选的文本挖掘工具，为多种文本挖掘方法集成，可以自动识别生物实体，又采用尖端的机器学习和深度学习技术来消除概念歧义，以提高准确性 [1]。

re 模块是 python 中处理匹配字符串所引入的一个模块，它包括正则表达式的所有功能。

ntlk 库是一个当下流行的,用于自然语言处理的 Python 库,在本次项目中我们主要利用其中的 sent_tokenize 函数进行分句、利用 word_tokenize 函数进行分词、利用 pos_tag 函数 wordnet 函数、WordNetLemmatizer 函数进行词性分析、词性还原、获取词性等操作。

3.2 研究方法中的核心思路

Gene 信息挖掘核心思路：主要利用 PubTator 结果文件的特点（上述已经提及），利用 re 库进行匹配，返回 dataframe 结果。

PTO mapping 中的核心思路：前面自己是采用分句匹配的方法，对 term name、synonym 进行分词，若都出现在同一个句子中，则作为匹配结果输出。后面学习了姚师兄提供的 mapping 算法，他是先将 abstract 进行分句，然后再分词，若 name、synonym 为词组则将句子和 name 转换为小写，直接用词组在句子中进行匹配；若为单个单词，则在分词之后的记过进行匹配。

基因与性状共现挖掘：将挖掘出来的 Gene 信息与文献中 TO 信息进行匹配，若 PTO 与 Gene 同时出现，则记录 pmid 号，gene ID，PTO 号等等。

3.3 本文的方法部分与课堂讲授内容的联系和区别/或补充

联系：夏老师上课讲授过 PubTator 获取实体信息的方法，并且还提供了 shell 脚本供大家使用。夏老师课上还提供了获取 PMID 号的方法，即利用 edirect 包中的 esearch 命令获取。关于 Gene 实体的挖掘，这一部分的代码是由我们自己完成。接着是 PTO 的挖掘，感谢姚师兄提供代码参考，以 1554 个 TO、Terms，这里的区别在于我采用的是硬匹配算法，本想参考姚师兄的，但由于之前提取的摘要信息格式不同，无法得出最终的结果文件。最后 Gene-PTO 共现挖掘，也是做一次 mapping 即可。

4 算法实践和代码编写要求

4.1 任务描述

首先利用 PubTator 工具对水稻文献进行实体标注，接着主要利用 python 中的 re 模块进行正则匹配获得 Gene 实体信息。接着对夏老师提供的 1554 个 TO Terms 进行处理，便于之后的 mapping。接着利用 python 中的 ntlk 包进行词性分析、词性还原、等方式进行匹配，最终与之前得到的’ Gene ‘信息进行共现挖掘，并通过 cytoscape 软件画出网络图。

4.2 实验设计

基因标注：在 PubTator 中按照下载的 PMID 号进行检索，利用 API 及 shell 脚本将全部文件摘要的注释信息下载。

Gene 信息挖掘：由于 PubTator 结果文件格式固定,其格式为 <PMID>|t|<title><PMID>|a|<abstract> 后面接着实体信息。因此我们可以根据这一特点对实体信息进行提取。我们编写 python 脚本，主要利用 re 模块中的 re.match 函数，最终我们得到一个只包含 Gene 实体信息的数据框。

PTO mapping：由于夏老师提供的 TO Term 信息由于是按行储存的文本信息，需要按行遍历文件内容。每个 TO Term 包括 “id”、“name”、“synonym” ，这一步我们利用姚师兄提供代码来对 TO term 文件进行处理，利用正则表达式匹配出我们需要的内容，将结果组成一个字典，字典的 key 是 id 信息，value 是相应的 name 和 synonym 信息。在上一步 Gene 信息挖掘中，我们不仅获得了包含 Gene 实体信息的数据框，我们还生成了一个包含有 PMID、title 以及 abstract 的列表。接着比对这一步，我最初的想法是只

将 abstract 分句，然后再与处理之后的 TO Term 信息进行 mapping，但结果并不好，会出现许多假阳性、假阴性的结果，例如缩写名称的广泛匹配，一些停用词的使用等等问题。后面学习了姚师兄写的代码，他首先将 abstract 中的文本信息进行了处理，利用 nltk 包里的词性分析、词性还原、提取词根词缀等方式进行匹配。

PTO-Gene 共现挖掘：由于 PTO 信息与文献摘要进行匹配的时候，一篇文献可能出现 TO 多次，因此需要先去重，然后再与挖掘的 Gene 信息进行匹配。

4.3 某些关键代码

```
参考代码链接：https://github.com/passion-web/NLP
#基因实体提取
for line in txt:
        if re.match(r'^\d{8}\|t\|', line):
            sublist = []
            head = re.match(r'^\d{8}\|t\|', line).group()
            pmid = head.rstrip('\|t\|')
            sublist.append(pmid)
            title = line.lstrip(head).rstrip('\n')
            sublist.append(title)
            text = line.lstrip(head).rstrip('\n') + ' '
            textout += line
        if re.match(r'^\d{8}\|a\|', line):
            head = re.match(r'^\d{8}\|a\|', line).group()
            #print(head)
            text += line.lstrip(head)
            textout += line + '\n'
            abstract = line.lstrip(head).rstrip('\n')
            sublist.append(abstract)
            alllist.append(sublist)
        if re.match(r'^\d{8}\t', line):
            data.append(line.rstrip().split('\t'))
    df = pd.DataFrame(data)
    df_gene = df.loc[df[4] == 'Gene']

参考代码链接：https://github.com/hongdoubao0315/PTO-Gene-
#硬匹配算法
for ta in alllist:
    count += 1
    pmid = ta[0]
    abs = ta[2]
    sent_list = [i for i in sent_tokenize(abs)]
    for sent in sent_list:
        for term,name_list in opt_dic1.items():
            for name in name_list:
                if name in sent:
                    wf.write('{0}\t{1}\t{2}\t{3}\n'. \
                             format(term, sent, pmid, '|'.join(name_list)))
                    count_result += 1

#利用Python画网络图
import pandas as pd
import networkx as nx
import matplotlib.pyplot as plt
import matplotlib.colors as colors
import matplotlib.cm as cmx
data = []
edges = []
```

```
with open('cyto1.txt','r') as fin:
    for line in fin:
        linelist = line.rstrip('\n').split(',TO:')
        sources = linelist[0].split(';')
        for source in sources:
            edges.append(source+'-'+linelist[1])
freq = pd.value_counts(edges)
data = []
for record in freq.iteritems():
    data.append([record[0].split('-')[0], record[0].split('-')[1], record[1]])
threshold = 0
plot = nx.Graph()
plt.figure(1, [9, 6], dpi=300)

cmap = colors.ListedColormap(['#FF300E', '#FF6B01', '#FFE500', '#7FF200', '#04CD65',
                              '#006AE0', '#7A14F3'])
cNorm = colors.BoundaryNorm([0, 1, 2, 3, 4, 5, 6, 20], cmap.N)
# cNorm = colors.Normalize(vmin=0, vmax=1)
scalarMap = cmx.ScalarMappable(norm=cNorm, cmap=cmap)
colorList = []
for record in data:
        plot.add_edge(record[0], record[1], weight=record[2])
        colorVal = scalarMap.to_rgba(record[2])
        colorList.append(colorVal)
filtered = [(u, v) for (u, v, d) in plot.edges(data=True) if d['weight'] > threshold]
pos = nx.random_layout(plot)
nx.draw_networkx_nodes(plot, pos, node_size=0.5, node_color='#99CCFF', alpha=0.7)
nx.draw_networkx_edges(plot, pos, edgelist=filtered, width=0.1, edge_color=colorList)
plt.colorbar(scalarMap)
plt.axis('off')
plt.savefig("weighted_graph.svg", format='svg', dpi=300)  # save as svg
plt.show()
```

5 主要的生物信息学实验和实验结论

Gene 挖掘：我们总共下载了 18299 篇文献的摘要，而 PubTator 的结果文件种只有 12782 篇文献的实体信息。我们猜测，可能是因为有些摘要中的实体信息 PubTator 无法识别，而导致文献摘要丢失。在 12782 篇文献中，挖掘出含有 Gene 实体信息的文献共 928 篇，仅有约 1/10 的文献含有 Gene 实体。而且挖掘出的基因种类只有 1625 种，接着我们统计除 Gene 词频数，利用 wordcloud 进行了可视化处理，方便更加直观地看到 Gene 情况，如图 1 所示。

PTO 挖掘与 mapping：我们总共从夏老师提供的 TO Terms 种提取出了 1531 个 terms，包含了 3110 个名字及同义名。接着我们利用硬匹配算法以及姚师兄提供的算法进行 mapping，利用硬匹配算法匹配到了 84367 个句子包含有 TO，而当我们利用姚师兄的匹配算法是不知是出于什么原因始终无法运行，生成文件大小始终为 0. 因此最终我们只采用硬匹配算法。结果发现出现最多的 PTO 是与植物茎腐烂抗性，总共出现了 48 次，其次是与产量有关的性状，总共出现了 28 次。接着我们对 TO Terms 进行从大到小排序，如图 2 所示。

PTO-Gene 共现挖掘：接着我们将得到的 PTO 信息与 Gene 信息进行匹配，我们找出出现频数最高的前 3 对有意义的结果，并且根据 PTO 性状含义对结果进行标注。结果如下图所示。可以看到，在同一篇文献中共同出现的 PTO-Gene，其 PTO 与 Gene 的生物学含义大体上还是对应，具有一定的参考价值。

同时我们利用 Python 中的 Networks 库画出 PTO-Gene 的网络图便于直观的描述，如图 3 所示。

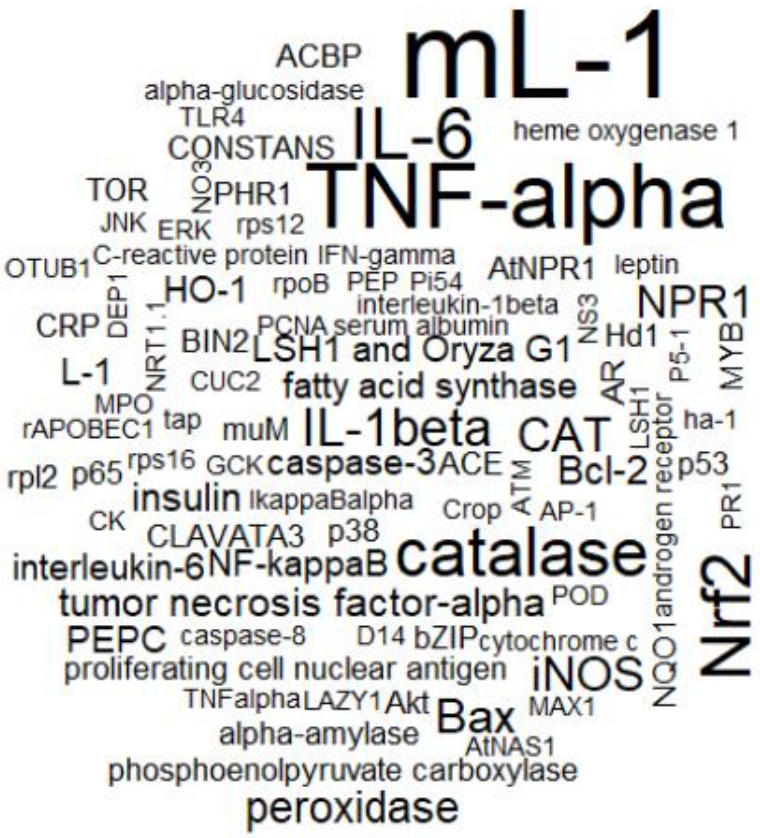


图 1: Gene 挖掘词云　图片来源：用 wordcloud 自绘

	pmid	PTO	term	time
5887	30497534	TO:0000323	stem rot disease resistance|CULMROTRS|SR	48
7935	29526465	TO:0000144	milled rice yield|milled rice ratio|MR	28
13785	30673437	TO:0000143	relative biomass|RB|relative biomass yield|RELBIOM	26
13787	30673437	TO:0000220	rice bug resistance|BUGRS|RB	26
13790	30673437	TO:0000356	brown spot disease resistance|BS	26
289	31521816	TO:0000260	bird damage resistance|BD	24
1709	29933132	TO:0000152	panicle number|NOP|NP|number of effective tiller per plant|...	24
7647	31234788	TO:0000424	brown planthopper resistance|BPH|BPHRS	24
10256	31035917	TO:0000121	ufra damage|U	24
11197	30387038	TO:0000294	leaf sheath auricle color|AC|AUCL	24

图 2: PTO 挖掘结果部分展示　图片来源：用 R 统计后截图

表 1: 部分 PTO-Gene 共现结果

PTO AND Gene	生物学意义	Time
TO：0000019 LSH1	苗高——以植物色素依赖性方式介导幼苗发育的光调节	13
TO：0000335 LSH1	苗的性状——以植物色素依赖性方式介导幼苗发育的光调节	13
TO：0000152 NPR1	穗数——水杨酸（SA）介导的系统获得性抗性（SAR）途径的关键调节因子	9

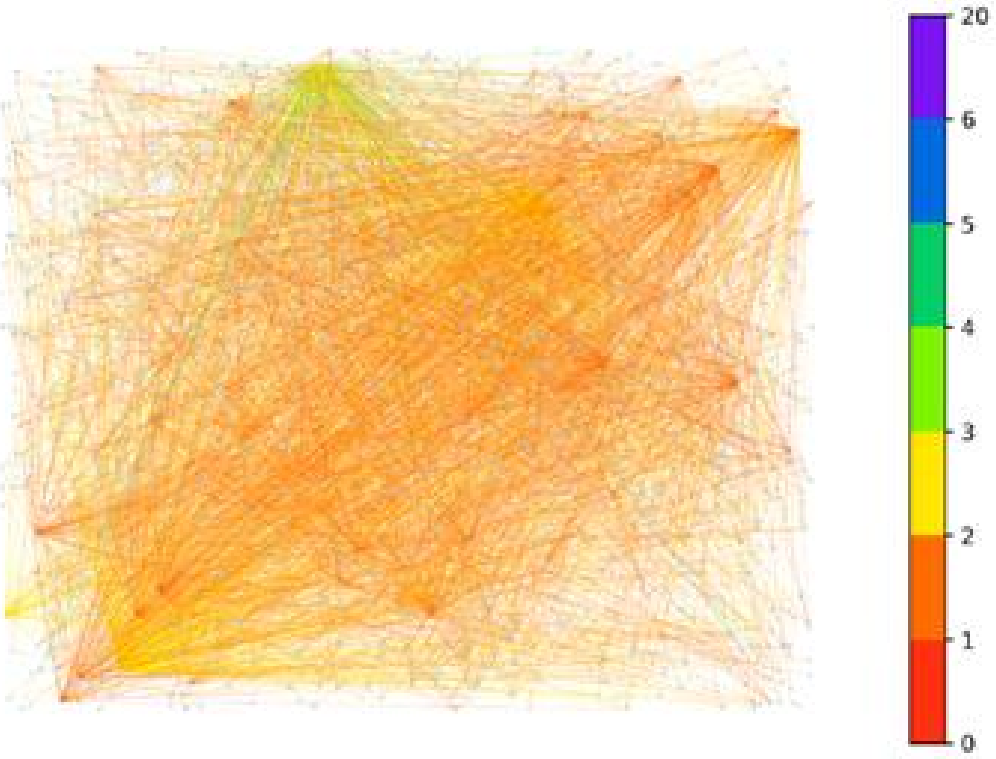


图 3: PTO-Gene 共现挖掘网络图 图片来源：用 python 库自绘

6 后记

6.1 课程论文构思和撰写过程

首先从 PubTator 获取实体信息开始说起，这里最初老师提供的脚本始终运行不了，始终提示非法格式，后面发现是由于不同系统 PMID.txt 文件会出现换行符错误，导致链接失效，有以下两种解决方法，第一种利用 vim 命令进入 pmid 的 txt 文件中，在命令行中输入 set fileformat = unix 即可。第二种方法是下载 dos2unix 包，利用 dos2unix file 命令即可解决。

再挖去 Gene 实体信息的时候，PubTator 会出现许多不一样的问题。例如：同一篇文献种 PubTator 挖出 PIN 和 PIN1 两种基因，其实它们是一种基因。还有你会发现结果中显示出现最多的基因是 mL-1 基因，我一开始还没反应过来，于是去 NCBI 上查找这个基因与什么有关，但发现这个基因在植物中不存在，于是我提取含有 mL-1 基因的 PMID 号，从 PubMed 上搜索，然后发现摘要中的 mL-1 代表着单位，等等还有一些类似的问题。关于第一个问题的解决方式，我想到可以先对摘要进行处理，例如词性还原，去除停用词等等处理方式，但这里又引入了一个新的问题，一些基因的简称会与停用词重叠，所以这只是我自身的一个想法。关于第二个问题，我目前想的是可以根据这个单词的上下文词性进行分析，然后再来做提取。

在做 PTO 挖掘的时候，我最初的想法是采用硬匹配算法以及姚师兄提供的代码，在这里，我碰到了一个安装包失败的问题，也就是 nltk 包中有些函数无法使用。后来，在百度的帮助下，找到了解决方法，在 host 文本下添加一个 githubusercontent 的 ip 地址，即可下载成功。随后成功的做出了硬匹配算法的结果，

但采用姚师兄的匹配方式，虽然说没有报错，但始终无法运行，在这里我猜测是不是由于我之前对于提取摘要的格式与姚师兄不同，所以导致了错误，这一点我仅是猜测。所以下一步的共现挖掘，我只能采用硬匹配的结果了。

基因与性状共现挖掘：最初的想法是如果基因与性状出现在同一句话里，则认为他们两者存在关联，后面发现，可以通过统计基因与性状在同一篇摘要中出现的结果，并绘制出网络图也可以作为分析依据。这里又回到了最初的 Gene 实体信息挖掘的问题了，你会发现，那个被 PubTator 误认为是基因的 'mL-1' 在挖掘结果中出现频率挺高的。所以说一个好的文本挖掘工具尤为重要。

6.2 所参考主要资源

S1. 本次项目代码链接. `https://github.com/passion-web/NLP`

S2. 参考代码链接. `https://github.com/hongdoubao0315/PTO-Gene-`

6.3 代码撰写的构思和体会

本次实验最初认为代码难度应该挺低的，认为就做两次 mapping 就好了，等真正到了自己操作的时候就会发现挺难的。并且其中对于 TO Terms 数据还要先做预处理，所以总的来说难度还是挺大的。最初的 Gene 实体信息提取这一步代码是自己写的，到后面利用姚师兄匹配算法的时候，发现可能是由于两者格式不兼容，所以导致运行不了，所以只能采用硬匹配算法，这是比较可惜的，最后还是感谢姚师兄提供的代码。

6.4 生物信息学实验设计的构思和体会

当拿到这个题目的时候，就询问了一下刘同学，感谢刘同学提供的思路帮助。这个项目思路还是比较简单的，两次 mapping 之后就可以得到基因与性状共现结果，不过中间一些数据处理还是比较复杂的。做完本次实验，自己还是获益匪浅的，一开始不明白自然语言处理是啥，现在已经初步了解了生物文本挖掘的一些知识。最后还是感谢夏老师以及夏老师的学生给我们项目提供的帮助。

参考文献

[1] C.H. Wei, K. Hung-Yu, and Z. Lu. Pubtator: a web-based text mining tool for assisting biocuration. *Nucleic Acids Research*, (W1):W518–W522, 2013.

5.2 赵柯韦、黄奇楠《水稻基因与性状的共句显示》

为了挖掘和发现水稻基因和性状之间的联系，以及在先前的实验中对这一关系的研究。我们尝试从和水稻相关的文献中提取出与水稻性状和基因相关的信息和数据，并且通过网络建模、依存树和词云等方法对收集和整理到的数据进行分析并从其中提取出具有一定价值和意义的信息。

课程论文 GitHub 网址：`https://github.com/Allen-ZKW/NLP_HZAU/tree/term`

水稻基因与性状的共句显示

赵柯韦 [1], 黄奇楠 [2]

[1] 华中农业大学信息学院，430070，武汉，湖北，中国
[2] 华中农业大学信息学院，430070，武汉，湖北，中国

摘要

为了挖掘和发现水稻基因和性状之间的联系，以及在先前的实验中对这一关系的研究。我们尝试从和水稻相关的文献中提取出与水稻性状和基因相关的信息和数据，并且通过网络建模、依存树和词云等方法对收集和整理到的数据进行分析并从其中提取出具有一定价值和意义的信息。

关键词: 水稻，基因，性状，共句显示，词云，依存树，网络

1 课题概况

本项目难度适中，而且涉及我们相对较为感兴趣的水稻的研究领域，在协商之下我们选择了这个题目来撰写课程论文。本文致力于挖掘过去多年研究中所涉及的对于水稻基因与性状的关系信息，预计可以基于挖掘出来的信息去构建一个“性状–基因”和多对多网络。

2 数据

本项目的文献原始数据是通过 Pubmed 所提供的 edirect 工具检索并下载所有与水稻 (Oryza sativa L.) 相关的英文文献摘要，并保存为 json 格式。将 PMID 从原始数据中提取出来，并保存为 txt 文件，将该文本文件作为 Pubtator 脚本的输入，批量下载 Pubtator 数据并保存为 txt 格式。

本项目的性状原始数据由老师所提供的 RTO 形状本体数据提供，下载后保存为 txt 格式。

3 研究方法

3.1 研究方法的算法背景，与其他方法的联系与区别

在本项目中所主要使用的算法是依存树分析算法，该算法会对我们提取出的语句进行分词，并且根据分词结果和训练结果为该语句构建依存树，依存树中的每个节点代表着一个词或者说是句子组分，而依存树中的边则表示着词和词之间的依存关系（控制词，依存词）。这种分析方式有利于我们提取句子主干和明确基因和形状在这个语句中的性质和交互关系。

实现相同的功能我们同样可以使用 CRF 或者逻辑回归去实现类似的功能。但是，在该项目中这两种算法我们认为都不如依存树算法合适，首先，逻辑回归算法和 CRF 所需要的训练数据需要手动注释，而受限于我们的知识水平和所接触的数据广度，我们很难做出精确且有价值的训练数据集。其次，逻辑回归和 CRF 的算法核心和依存树不同，二者本质上都是一种分类算法，并不会生成词和词之间的关系，而依存树所构建的依存关系和我们在实验中所需求的“性状–基因”关系相对更加契合

3.2 研究方法中的核心思路

在进行该项目时，我们的核心思路大体分为三个部分：数据的下载和整理-> 数据的提取-> 数据的分析与可视化。在数据的下载和整理的过程中尽可能多地同时获得相关的注释信息，并且将数据整理成相对更加有利于提取和分类的形式防止因为数据形式的问题导致数据提取的难度提高。在数据提取的过程中，我们需要尽可能的考虑到基因和性状所在语句中所可能出现的各种情况，并尽可能地覆盖所有结果。得到数据后，需要对数据进行分析和判断，这一过程中，我们应尽可能多的应用课堂上所学习的各种分析手法，多角度，多层次地体现结果的特征和性质。

3.3 本文的方法部分与课堂讲授内容的联系和区别/或补充

本项目中的方法部分大多数都参考自课堂上讲授的内容，其中在分句和依存树分析部分基于本项目具体情况做出了部分调整：

首先，由于 Pubtator 所给出的注释中包含了 GO 的 start 和 end，所以我们并没有直接使用相关模块中的分句语句，而是自己编写了脚本将标题和摘要合并并记录每一个句子（标题视为首句）的 start 和 end，用于方便后续的语句提取。

其次，根据依存树的结果，我们发现基因和性状之间的联系不能简单的使用课堂上所讲授的 nsubj-ROOT-dobj 这种简单的模型来反映。基因在很多情况下是作为语句中的同位语或者其它成分存在，除此之外，形状本体很多情况下是一个短语而非一个独立的单词，所以，我们在进行依存树分析的时候也进行了相关的调整来适应这个实验：将过滤条件中引入了除 nsuj 和 dobj 之外的 nsubjpass 来尽可能捕获更多的符合条件的语句；如上文所述，我们采取了将词组转化为单词来防止依存树分析将词组分解拆分，来实现整体的标注。

在词云的绘制中，不同于使用 R 语言处理数据并绘制图像。由于 R 语言包 wordcloud2 存在缺陷，我们无法使用这个包去绘制指定图形的词云。我们采取了使用 R 语言进行语料库的过滤、分析以及抽取高频率的词汇表。基于 R 语言的处理结果，再调用 python 中的 jieba 模块实现分词吗，并通过 wordcloud 包对分词的结果进行词云的绘制，最终得到了如本文中图 1 中所示的特定形状的词云。

4 算法实践和代码编写要求

4.1 任务描述

从原始文献摘要中提取同时含有 GO 和 TO 的语句，筛选出这些语句中可以体现明确的性状和基因的关系的语句。并对这些语句进行分析，将分析结果可视化，同时从中获得相关的生物学意义。

4.2 实验设计

共句表达的提取：由于我们所使用的数据来源于 pubtator，所以先天带有和基因相关的注释，我们首先删除了所有非基因注释而且合并了标题和摘要。其次，根据上文所述的分句方法（生成每句话的 start 和 end）可以很方便的提取出含有 GO 的语句。另一边，我们提取了 RTO 文件的 name 项和 synonym 项整理为一个 TO 字典。通过调用 re 包中的相关语句实现正则表达和匹配，并通过循环使字典中的每一项都和每一句含 GO 的语句进行匹配，并根据匹配结果筛选出同时含有 GO 和 TO 的语句。

依存树筛选：为了确定这些语句中基因和性状之间是存在某种联系，而非只是单纯的，恰好的出现在同一个语句中，我们对这些语句进行了依存树分析，并保存了依存树分析的完整结果。根据这些结果去判断删除还是保留这个句子，该筛选过程先由脚本进行较为严格的筛选，在删除的句子中部分由人工观察保留下来。

分析与可视化：将筛选出来的句子整合成为一个语料库，在进行数据清洗后建立一个词云去反应各种词语在这些语句中出现的频率。另一方面，通过 TO 标准字典将所有的 synonym 都转化回 name，然后基于 GO-TO 这一对关系去构建一个关系网络，并且通过调用相关的模块，计算该网络的包括中心性，连通性在内的一系列参数来反映该网络的性质。

其中较为复杂的是共句表达的提取，因为我们事先无法确定 TO 在句子中的形式以及出现的位置，所以我们只能对句子和 TOname 都进行了格式统一化，即删除所有标点符号同时将所有大写转化为小写后进行匹配。但这种相对粗暴的操作也带来了一系列问题，首先，有些对大小写有严格要求的 TO 被错误匹配到了小写的一般单词上，如 BY 和 An 就会出现这种问题。其次通过阅读 RTO 文档，我们发现了许多同义词或者说字典中的许多词是全集与子集的关系，比如，某句中出现了 molecular hydrogen sensitivity，在进行正则匹配的时候 hydrogen sensitivity 和 molecular hydrogen sensitivity 这两个对象都可以匹配，但是实际上应该只允许 molecular hydrogen sensitivity 正确匹配到该句。

除此之外，在依存树分析中，也出现了一系列未曾设想的问题。首先，由于 TO 往往是以词组的形式存在，在进行依存树分析的时候，依存树往往会将这个词组拆分然后分别标注。另外，通过观察未经处理的依存树结果，我们也发现了在部分语句中，依存树不能很好的识别基因实体，这导致了标注缺失的问题。我们尝试采用了代换的方法来解决这两个问题，即使用 TO1，TO2，TO3 · · · · · · 来代替语句中出现的所有 TO，使用 GO1，GO2，GO3 · · · · · · 来代替所有出现在语句中的 GO，以便于计算机有效识别。

基于获取的共句信息，我们调用 networkx 模块构建了一个性状与基因互作的网络，并将关系所出现的频次作为参数赋值给了代表该关系的边。基于研究的问题，我们选择了点度中心性和度分布这两个参数作为反应该图性质的主要参数。并且我们通过 Cytoscape 软件实现了该网络的可视化。

4.3 实验关键代码

```
#代码来源：赵科韦编写
#GO提取与文本分句
for row in f:
    if '|t|' in row:
        key = row.split('|t|')[0]
        title = row.split('|t|')[1].strip()+'␣'
        p_d[key] = {}
        p_d[key]['paper'] = title
        p_d[key]['annotation'] = []
    elif '|a|' in row:
        key = row.split('|a|')[0]
        abstract = row.split('|a|')[1].strip()+'␣'
        p_d[key]['paper'] = p_d[key]['paper'] + abstract
        start = [0]
        stop = []
        for i in range(len(p_d[key]['paper'])):
            if p_d[key]['paper'][i:i+2] == '.␣':
                stop.append(i+2)
                start.append(i+2)
        del start[-1]
        sentence = []
        for i in range(len(start)):
            sentence.append([start[i],stop[i]])
        p_d[key]['sentence'] = sentence
    elif row != '\n':
        note = row.strip().split('\t')
        if note[4] == 'Gene':
            p_d[key]['annotation'].append(note)
```

```
#代码来源：赵科韦编写
```

```
#TO匹配与过滤
for TO in dictionary:
    if TO not in special_TO:
        pure_TO = TO.translate(str.maketrans('', '', string.punctuation)).lower()
        pattern = re.compile('\\b' + pure_TO + '\\b')
        m = re.search(pattern,pure_sentence)
        if m != None:
            gts[key]['TO'].append(TO)
    else:
        pure_sentence = key.translate(str.maketrans('', '', string.punctuation))
        pure_TO = TO.translate(str.maketrans('', '', string.punctuation))
        pattern = re.compile('\\b' + pure_TO + '\\b')
        m = re.search(pattern,pure_sentence)
        if m != None:
            gts[key]['TO'].append(TO)
```

```
#代码来源：赵科韦编写
#依存分析
nlp = spacy.load('en_core_web_sm')
for key in gts.keys():
    sentence = key
    num =1
    for i in gts[key]['GO']:
        sentence = sentence.replace(i,'GO'+str(num))
    num = 1
    for i in gts[key]['TO']:
        sentence = sentence.replace(i,'TO'+str(num))
    word_pos_dict = {}
    doc = nlp(sentence)
    for token in doc:
        word_pos_dict[token.dep_] = token.text
    gts[key]['depency_result'] = word_pos_dict
```

5 主要的生物信息学实验和实验结论

5.1 依存关系分析结果

依存关系分析的结果较为不理想，通过观察依存树分析结果我们推测这种情况可能是由三方面原因所引起的：一、语句来源于科研论文，其语句句法的复杂性相对较高，依存关系的判断本来就难度较高；二、我们在实验中所使用的训练数据可能不能很好地处理生物文本信息，导致了在进行依存分析时，许多 TO 和 GO 出现了无法识别的情况；三、我们所使用的句子中标点符号数量和种类都偏多，这可能也是导致较差结果的原因。但是基于依存分析结果我们仍然获得了 41 个具有强烈相关关系（即作为主语和宾语）的性状基因对。分析这些关系可以看出，对于水稻的基因与形状的研究主要还是立足于农业的，并且主要集中在水稻的抗性和生产这两个方面，即水稻的不同基因可以带来怎样的有利于农业生产的性状，在我们的提取出的结果中涉及了干旱耐性，高盐耐性，一些氨基酸在水稻中的含量，水稻的生长发育等一系列性状。

5.2 网络模型可视化

从网络模型中观察可以看出，共句显示结果中所涉及的基因本体和性状本体数量较多，且大部分性状、基因可以通过共句关系构建一个规模相对较大的网络，只有少部分基因和性状是游离在这个大规模网络之外的子网。我们分析认为可能会有三种原因导致这种情况：1）确实存在部分性状和基因间存在较为独立的联系。2）此网络是基于过往的研究成果进行的信息挖掘，可能这些游离的网络与大型网络存在某种联系，

但现在未被发现或者没有在这些摘要中体现。3）pubtator 无法完全提取所有的 GO 数据，且 TO 源文件和匹配过程都存在遗漏的可能性。因此导致了部分关系没有被我们挖掘出来。

5.3 网络度分布可视化

根据对网络的度分布结果的假设检验和可视化可以看出，该网络的度分布与幂律分布较为相近，即少数节点拥有大量连接，而大多数节点只拥有少数连接，具有着严重的不均匀分布性 [1]。也就是说少数的 TO 和大量的 GO 存在联系，少数 GO 与大量的 TO 存在联系。这在一定程度上体现了在水稻中存在部分较为关键的基因这些基因可能同时调控着下游大量的性状，同时也存在着部分调控机制或作用机制较为复杂的性状，这些性状受到大量基因的调控。该网络同时具有鲁棒性和脆弱性，一方面对于随即故障的容错能力强，另一方面，如果关键节点受损，很可能破坏整个网路的功能。对于水稻这个系统来说，一方面其作为一个生物系统拥有一定的抗逆能力和适应能力，而另一方面如果水稻的关键基因，或者关键功能（如光合作用或者糖酵解过程）收到损伤，有可能会导致水稻的生长不良甚至死亡。

5.4 实验结果展示

以下是本次课程项目结果的部分图片展示 [1]：

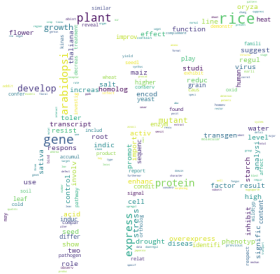


图 1: 水稻词云图

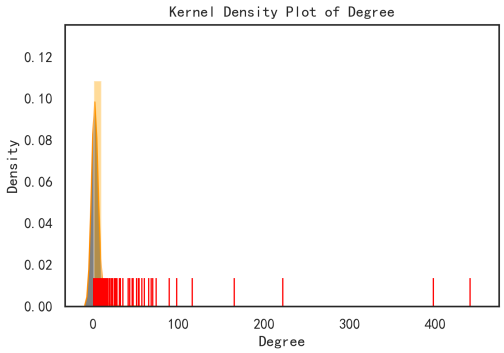


图 2: 网络度分布密度曲线图

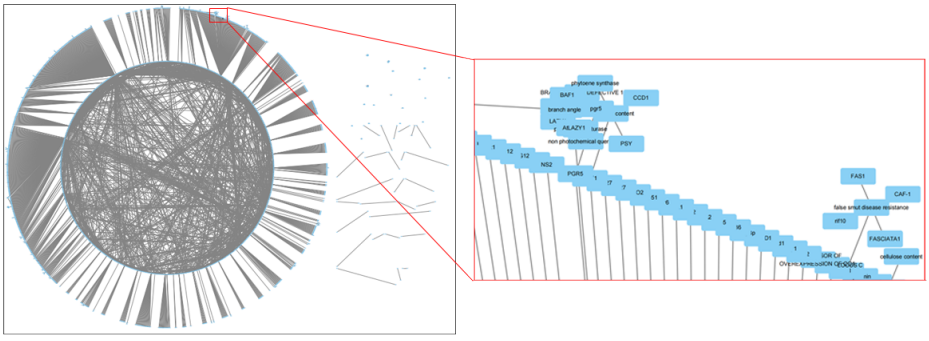


图 3: 形状本体与基因本体的共句关系网络

[1]图 1 在 R 中运用 wordcloud2 包实现绘制；图 2 在 python 中运用 matplotlib.pyplot 包绘制；图 3 利用 Cytoscape 绘制而成，再局部放大

本次实验除了利用图像对数据进行可视化外，我们还将网络度分布和词频进行了表格可视化处理，具体结果如下：

表 1: 网络度分布表格 (部分)

TO	GO	Freq
growth	mL-1	2
phenotype	pex5	1
heat	AtSIZ1	3
morphology	ROXY1/2	1
rat damage resistance	Exp.1	2

表 2: 词云频次表格 (部分)

Word	Freq
rice	908
gene	677
plant	556
express	487
arabidopsi	474
protein	413
develop	345
growth	318
stress	313

6 后记

6.1 课程论文构思和撰写过程

本项目主要侧重于数据的处理和分析上，在算法部分偏向于薄弱，因此在撰写论文时，我们也主要侧重于讲解我们在数据处理和在结果分析上所遇到的问题和经验。并且尽可能地使用更多的分析手法来从不同层次和不同的方向分析我们所获得的结果，同时对这些分析结果也尽可能地做出可视化和再分析。并根据这些得到的数据，寻找到其中相关的生物学知识和解释。

在论文结构上，我们并没有严格遵循我们进行实验时的时间顺序，而是根据本项目的逻辑关系，将各个步骤进行了整合和归类，然后，进行了论文的撰写。

6.2 所参考主要资源

本文所使用的性状本体信息来源于由夏静波老师所提供的 RTO 数据 [2]，所使用的摘要数据信息来源于由 Pubtator 下载的信息。在自然语言处理中所应用的主要知识和思路主要参考自夏老师上课时所使用的 PPT 和课堂教学内容 [3],[4]。在网络构建和分析的部分主要参考了马彬广老师在系统与合成生物学上所使用的 PPT 和教学内容。

6.3 代码撰写的构思和体会

在最初的代码思路中，我们本来是想要基于 NCBI 数据库下载所有关于水稻的基因，并进行手动匹配的。不过在后续的处理中，我们发现这样的做法容易存在大量的错配和未匹配。于是，在 GO 的提取上我们直接使用了 Pubtator 的注释结果，并改变了原先的分句方法以适配 Pubtator 的注释格式。

在 TO 的匹配上，由于我们抹除了语句和 TO 的所有格式和标点，这使得结果中出现了两个难以解释的峰值数据 “BY” 和 “An”，我们在之后选择对这部分的 TO 做特殊处理，即严格匹配大小写和格式。除此之外，由于我们之前的数据储存格式是 sentence+TO+GO 这种格式，其中记录了大量无用且重复的信息，之后我们将数据储存格式改为 sentence+TO_set+GO_set，有效节省了储存空间，也方便了后续的过滤。

在共句的过滤中，我们主要针对三种情况进行过滤：1）包含型：在同一句中，一个 GO 被另一个 GO 完全包含。2）特殊型：如上文所述，对特殊型 TO 进行更加严格的匹配。3）空值型：在进行完上两步过滤之后，清除含有空列表的项目。本来在计划中还存在着依存信息过滤，但是由于上文中所示的原因，该部分并未在本文中应用和展示。

在网络搭建中，我们认为一个 TO 项目的 name 和 synonym 不应该在网络中处于两个不同的节点，所以我们调整了对于 TO 的处理，让其不仅产生一个 TO 字典，同时也会产生一个标准字典，即 name/synonym :name。来实现输入后返回该输入的标准命名。

本项目的代码难度偏低，但涉及问题和细节较多，不仅要求我们在初次编写代码时就尽可能思考全面，而且需要我们在撰写代码时不断改进原先的代码以满足当前发现的问题。

6.4 生物信息学实验设计的构思和体会

根据项目要求，我们在初步构思该实验的时候仅仅考虑到了数据的提取和共句信息的提取。但是，随着学习的深入，我们逐渐意识到仅仅获取结果并不能算是一个完整的研究过程，这个结果如果不进行分析和处理，我们是很难从结果中提取出具有生物学意义的结论。在思考如何分析数据时，一方面，我们认为在本课程上所学习的词云方法是一种直观且具体的方法去反应这些句子的高频词汇；另一方面，共句表达本质上是一种本体与本体之间的联系，所以我们认为针对这些共句关系构建一个网络，并且将关系出现的频率作为属性赋值给网络是一种很好的数据处理方法。

我们组的实验设计是随着实验的推进而不断发生改变的，有一些新内容（如词云、网络）被添加到实验设计中，也有一些内容（依存树）由于结果较差，虽然仍然进行了实验，但没有被包含在结果之中。这些缺陷和后期完善的部分将会成为我们未来设计实验时的教训和经验。

6.5 人员分工

在本项目中，赵柯韦和黄奇楠采取了独立编写的同时相互交流和参考的方式进行，两人均以不同的方式实现了项目目标。论文撰写也由两个人共同完成。

赵柯韦：实验设计；代码编写；论文的内容书写，论文核查与修改。

黄奇楠：实验设计；代码编写；论文的格式书写，论文核查与修改。

7 附录

S1. 2021 年课程网页. `https://hzaubionlp.com/course-bionlp-and-kd/`

S2. 课程论文可以参考使用的基础代码，可参考 GitHub 页面, `https://github.com/bionlp-hzau/`

S3. 课程论文所使用的全部代码、数据及详细结果，可参考 GitHub 页面，`https://github.com/Allen-ZKW/NLP_HZAU/tree/term`

参考文献

[1] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[2] Xinzhi Yao, Jingbo Xia, Kaiyin Zhou. Tomapping. `https://https://github.com/bionlp-hzau/TOMapping/`.

[3] Jingbo Xia. Tutorial4wordcloud-basic. `https://https://github.com/bionlp-hzau/Tutorial4WordCloud-Basic/`.

[4] Kaiyin Zhou, Jingbo Xia. tutorial for dependency tree. `https://github.com/bionlp-hzau/tutorial4dependencytree/`.

5.3 陶芳婷、黄婷婷《水稻基因-性状的挖掘》

水稻是全球约半数人口的主食，也是我国的主要粮食作物，拥有着悠久的种植历史。水稻也因为它极大的研究价值一直以来都是农业科学家的研究热点。目前国内外科学家关于水稻有着比较详尽的研究，也获得了很多的成就。为了进一步了解水稻基因和性状的关系，文章对两万篇水稻相关文献进行数据挖掘。通过水稻基因和性状本体的共句显示，发现水稻的很多性状受到基因 N 的影响，以及科学家们水稻性状的研究热点是其生长等。

课程论文 GitHub 网址：`https://github.com/huangtingting123/`

水稻基因—性状的挖掘

陶芳婷 [1], 黄婷婷 [1]

[1] 华中农业大学信息学院，430070，武汉，湖北，中国

摘要

水稻是全球约半数人口的主食，也是我国的主要粮食作物，拥有着悠久的种植历史。水稻也因为它极大的研究价值一直以来都是农业科学家的研究热点。目前国内外科学家关于水稻有着比较详尽的研究，也获得了很多的成就。为了进一步了解水稻基因和性状的关系，文章对两万篇水稻相关文献进行数据挖掘。通过水稻基因和性状本体的共句显示，发现水稻的很多性状受到基因 N 的影响，以及科学家们水稻性状的研究热点是其生长等。

关键词：水稻，基因，性状

1 课题概况

关于选修这门课程的目的，我们知道，21 世纪是数据大爆发的世纪，如何进行处理是一个亟待解决的问题。希望能通过学习《自然语言处理与知识发现》这门课，掌握相关的数据处理方法，提升自己的专业素养。关于选取当前课程论文题目的原因，水稻是我们一日三餐所必需，对于我们来说是比较熟悉的物种，在进行数据挖掘分析相关的生物学意义会比较轻松。

文章希望通过对水稻基因-性状的大数据挖掘，获取当前水稻研究的热点以及已研究得到的水稻基因-性状的潜在关系。

2 数据

2.1 提取文献摘要

本文利用 PubTator 提供的 PubTatordb 数据库 [1]，根据水稻的 TAXID（4530）对 species 表进行筛选，得到水稻文献的 PMID，并取前 20000 篇文献。随后利用 edirect 根据这些 PMID 提取文献信息，将其转化为 xml 格式，并通过 edirect 提供的 xtract 直接获得文献摘要 [2]。

2.2 获取水稻性状本体

根据网站 https://github.com/bionlp-hzau/TOMapping 中的 RTO-1.0.obo 获取水稻性状本体。

2.3 获取基因本体

PubTatordb 数据库中的 gene 表可以通过已知的 PMID 筛选得到所有文献中所有基因本体的 EntrezID。下载 NCBI 上的 gene_info 文件，首先通过水稻 TAXID 筛选出水稻所有的基因信息，随后利用脚本将 PubTator 得到的 EntrezID 转化为 symbol ID，便于后续研究。

2.4 数据规模和格式

所有的数据均为 txt 文本格式，其中文献数据包含 PMID 和摘要两列数据; 文献 20000 篇，基因本体 3365 个，性状本体 2050 个。

3 研究方法

3.1 研究方法的算法背景，与其他方法的联系与区别

PubTator 是一个基于 Web 的系统，提供 PubMed 摘要和 PMC 全文文章中的基因和突变等生物医学概念的自动注释。edirect 全名为 Entrez Direct，包含了一组各司其职的工具和脚本，通过 Linux 系统下的管道功能，联合使用这些命令行工具，可在服务器上高效的进行 Entrez 的检索，抓取，过滤，排序等操作。

3.2 研究方法中的核心思路

本项目利用 PubTator，edirect 这两个工具来提取文献摘要、基因本体、性状本体等项目数据。编写 python 脚本，根据句号分割摘要，利用通配符匹配的方式来找寻基因和性状之间的共句显示。

根据得到的结果文件进行进一步的数据处理，通过相应的 shell 命令以及所编写的简单的 perl 脚本对数据进行处理，并进行相应的可视化。

3.3 本文的方法部分与课堂讲授内容的联系和区别/或补充

在数据搜集方面，本文在数据搜集方面借鉴了课堂讲授的 PubTator 和 edirect 的使用方式。但为了快速获取基因实体，用 PubTatordb 本地的 R 数据库代替了传统的通过 url 获取信息的方式。直接利用 PMID 而不是关键词，通过 edirect 获取文献信息。

在可视化方面，一部分的可视化借鉴了语料库学习的内容。还有一部分的可视化是根据系统生物学课上的学习内容以及自己平时的实践经验进行绘制。

4 算法实践和代码编写要求

4.1 任务描述

4.1.1 数据获取

本文数据包括文献摘要、基因本体、性状本体这三个部分。其中性状本体根据课程网页的数据可以直接通过其 NAME 来获得。

PubTatordb，提供 chemical，disease，gene，mutation，species 五种表格查询。其中 species 表格包含 PMID、TAXID、MENTIONS、RESOURCE 四列。gene 表格包含 PMID、ENTREZID、MENTIONS、RESOURCE 四列。观察这两个表格，我们可以发现通过水稻 TAXID 获取相关文献，并根据 PMID 获取基因实体的 EntrezID 是可行的，本地数据库同时满足了批量快速的提取要求。虽然不能从 PubTator 中直接得到文献摘要，但 edirect 提供了由 PMID 提取文献的方式。

4.1.2 共句显示

此过程主要使用了 python 脚本来进行处理。由于文献摘要的格式要求，几乎所有的文献都采取句号来分割语句。我们可以根据句号来将摘要文本分割成多个句子，并对所有的句子进行遍历，随后遍历性状和

基因本体，通过 re.search 判断每个句子是否同时与基因和性状相匹配。最后结果输出 5 列，依次代表基因本体、性状本体、共句次数、所出现的文献数量、文献 PMID（用；分隔）、共句的句子文本。

经过上述设计，我们发现多层循环语句会造成脚本运行的缓慢，因此我们可以将 20000 篇文献进行分割，多线程同时运行。编写另一个 python 脚本——merge.py，对所有的结果进行合并。

4.1.3 可视化

该过程主要对共句显示得到的文件进行可视化方面的相应数据处理。通过得到的文件，结合自己所学的生物学知识，经过数据处理挖掘其可能存在的生物学关联及生物学意义。

4.2 实验设计

4.2.1 数据预理

在 github 上查询 PubTatordb，在 R 中导入该数据库。

NCBI 上查询利用水稻的 TAXID 为 4530，根据 PubTator 提供的 select 函数提取 PMID，取前 20000 篇文献。利用 edirect 根据 PMID 提取摘要，输出格式为 XML，直接使用 xtract，根据摘要和 PMID 的标签，获取摘要文本。

根据这些 PMID，从 gene 表中找出 gene 实体（EntrezID 格式）。下载 NCBI 的 gene_info 文件，根据 taxid 筛选水稻所有的基因名称，然后遍历 EntrezID 文件，获取对应的 symbol ID，得到所需基因实体文件。

4.2.2 共句显示

编写 sentence.py 输入基因本体文件 gene_sym.txt, 性状文件 rice_trait.txt, 文献摘要 abstract.txt 文献。

遍历 gene_sym.txt，rice_trait.txt 文件将本体存储到两个数组内。

遍历 all_abstract.txt，将每一篇摘要利用 split 函数对‘. ’进行分隔（句号后包含一个空格，避免用.进行连接的词语的干扰）。遍历基因数组，利用 re.search 判断，如果该句匹配基因，则遍历性状数组，再次判断。同时匹配，则将该性状和基因用‘;’连接，作为键，将该句子、文献 PMID 存入元组中。

关于 re.search 注意事项，考虑到词语在句子中的位置会影响到判断，因此我们需要对句中、句末、句首的这三种情况都进行判断。同时还要加上 re.I 参数，排除大小写的影响。

4.2.3 文件合并

将 all_abstract.txt 文件利用 shell split 命令分割成 10 个文件，每个文件包含了 2000 篇摘要。对这 10 个文件分别运行 sentence.py，多线程并行运行。

编写 merge.py，将 10 个结果文件进行合并。

2000 个摘要作为输入文件运行 sentence.py 需要 60 分钟左右的时间，我们每次运行 5 个进程，获得最终结果文件需要 2 小时的时间。

4.2.4 数据可视化

首先经过排序及相关的命令得到共句显示数量排名前 3 的“基因-性状”所在句子汇总，得到相关词及其频率的表格，并分别进行词云的可视化。然后，通过编写 perl 代码及相应的命令得到在 20000 篇文献中提及数量前 10 的基因及性状，以及它们出现在文献里的相应次数，得到表格。最后基于上一步骤产生的文

件，选择其排名前 5 的基因/性状，通过编写代码得到其所共文献数量大于 1 的性状/基因及其相应的共文献数量，输出为 Highcharts 网站绘制桑基图的相关文件。

4.3 某些关键代码

```
代码来源：：https://github.com/MAMC-DCI/pubtatordb
代码目的：查询pubtatordb，获取基因实体
gene_table<-pt_select(
db_con,
"gene",
columns=c("ENTREZID"),#只获取ENTREZID
keys=PMIDs,#前面通过species表格获取的PMID向量
keytape="PMID" ,#关键字为PMID
Limit=Inf
)
```

```
代码来源：自己编写
代码目的：根据PMID，利用edirect获取摘要
cat PMID_ac|while read line;do efetch -db pubmed -id ${line} -format xml|xtract -pattern PubmedArticle -
    element MedlineCitation/PMID -block Abstract -sep "␣" -element AbstractText >>abstract_ac.txt;done
```

```
代码来源：自己编写
代码目的：查询共句情况
bstract_dic[lis[0]]=lis[1]
abst=lis[1].split('.␣')#根据'. '进行分隔，.后面包含空格，用于区别用.连接的词语
for i in abst: #遍历摘要
for gene_name in gene_array:#遍历gene_array
#通配符匹配，分别包括句首，句中，句末三种情况，由于之前使用'. '分割，因此每个摘要的最后一个句子的'.'没有
    去除，句末需要考虑两种情况
if      re.search('^'+gene_name+'[\s\']|\s'+gene_name+'[\s.\']|\s'+gene_name+'$',i, re.I):
for trait_name in trait_array: //遍历trait_array
#通配符匹配
if re.search('^'+trait_name+'[\s\']|\s'+trait_name+'[\s.\']|\s'+trait_name+'$',i, re.I):

dic_key=gene_name+";"+trait_name #构建组合
if(out_dic.get(dic_key)): #判断该键是否存在
.....#后面代码省略
```

5 主要的生物信息学实验和实验结论

最后一共有 478 组基因性状存在共句显示，涉及到 225 个基因和 108 个性状。根据共句显示得到的句子，使用 Cytoscape 绘制基因-性状的网络关系图，如图1所示。

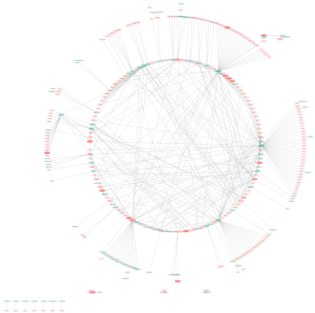


图 1: 网络图。图片来源：自绘，工具：Cytoscape。

其中，红色节点代表基因本体，绿色节点代表性状本体。如果性状和基因之间存在共句显示，则两个节点间存在连接线。同时，节点存在的共句数目越多，则颜色越深。

根据共句显示文件，得到频率排名前三的基因-性状所在的句子汇总，统计其关联词汇及出现次数，得到表1。

表 1: 提及频率前三的基因-性状关联词汇及出现次数

N-growth		N-chemical		ACT-development	
word	freq	word	freq	word	freq
growth	44	chemical	12	act	12
rice	26	rice	9	development	11
soil	17	fertilization	4	expression	6
contents	7	nitrogen	4	may	6
nitrate	7	soil	4	rice	6
plant	7	balanced	3	arabidopsis	4
application	6	fertilizer	3	gamyb	4
different	6	increased	3	genes	4
nitrogen	6	plant	3	udt	4
stages	6	respectively	3	plant	3

根据表1，我们可以得到在研究 N-growth、N-chemical、ACT-development 时，主要是研究土壤、营养物等因素对其造成的影响，并通过收获的稻米得到进一步验证。

基于形成表1过程中产生的数据，对 N-growth 所在的句子相关词汇及出现次数绘制词云，得到图2。

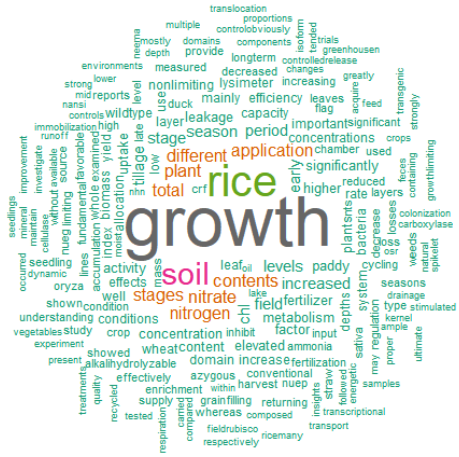


图 2: 云图。图片来源：自绘，工具：Rstudio。

然后筛选共句显示文件，基因/性状所在文献数量，得到表2。

根据表2，我们可以了解到，关于水稻基因的研究，科学家对基因 N 的研究最多，其次是基因 ACT 和基因 PEROXIDASE；关于水稻性状的研究，科学家对性状 development 的研究最多，其次是性状 growth 和性状 phenotype，这也比较符合现实意义。水稻作为全球约半数人口的主食，了解水稻生长情况并进而基于此提高其产量一直都是研究热点。

表 2: 相关基因和性状在文献中出现的次数

gene	numbers(gene in abstracts)	trait	numbers(trait in abstracts)
N	95	development	101
ACT	37	growth	98
PEROXIDASE	21	phenotype	35
SSR	19	disease	29
NAC	17	drought	23
SI	16	heat	17
ACHE	14	cold	16
NPR1	12	virus	14
ATP6	12	photoperiod	14
CAT	10	quality	12

进而在表2的基础上，对共句显示文件进行进一步的处理，选择其排名前 5 的基因/性状，得到其所共文献数量大于 1 的性状/基因及其相应的共文献数量数据，然后通过 Highcharts 网站绘制桑基图，得到图3。

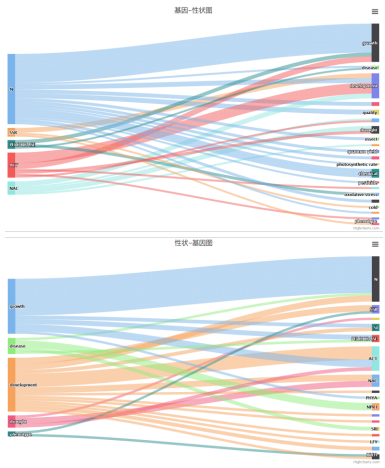


图 3: 桑基图。图片来源：自绘，工具：Highcharts。

根据图3，我们可以了解到基因 N、ACT、PEROXIDASE、SSR、NAC 对哪些性状起作用，性状 development、growth、phenotype、disease、drought 受到哪些基因的调控。根据基因-性状图，可以看出基因 N 作用的性状最多，可以推测它是水稻基因中的一个关键基因。根据性状-基因图，可以看出性状 development、growh 受到很多基因的调控，说明水稻的生长受到很多基因的影响。纵览全图，也可以得到科学家的研究热点主要集中在 N-growth 上，这不仅说明性状 growth 是一个研究热点，也说明了基因 N 对性状 growth 的重要性。

6 后记

6.1 课程论文构思和撰写过程

关于课程论文的构思，本次课程项目重点在于通过代码找出共句显示，以及结果文件的可视化。我们突出描述了数据搜集和数据处理的代码处理过程，并采用表格展现了一系列共句数量的统计结果，利用图片展现了基因-性状关联的多样性，充分挖掘了课程项目和课堂教学之间的联系，并详细展示了学习过程中的探索过程和相关体会。

关于其撰写过程，每个人负责其代码实现的内容，并经过汇总，使用 LaTeX 进行文章的排版。因为是第一次使用这个软件，遇到了很多挫折。本来是打算使用群里分享的 TeXworks 软件，但是在构建并查看

文件的时候一直报错，经过相关搜索，下载安装了 TexLive 和 TeXstudio，并在 TeXstudio 进行编译。在编译过程中，也遇到了一些问题，图片并不能置于我想要的段落下，文章中出现”_” 会报错，参考文献的引用等。

6.2 所参考主要资源

LaTeX 的入门文档：https://www.kancloud.cn/thinkphp/latex/41808
LaTeX 使用模板：https://hzaubionlp.files.wordpress.com/2021/04/template4coursepaper-bionlp2021version.zip
Edirect 相关代码：https://www.ncbi.nlm.nih.gov/books/NBK179288/
pubtatordb 使用参考：https://github.com/MAMC-DCI/pubtatordb

6.3 代码撰写的构思和体会

NCBI 上文献摘要的下载一次性只能提供 10000 篇的下载量，并不支持更多的下载。对于 PubTator，如果采取的课堂所授的方式，通过 url 下载，碍于脚本睡眠的时间无法实现大批量的快速提取。因此我们在这里尝试了本地数据库的方式。但是本地数据库获取的数据非常的单一（例如基因只有 EntrezID），其资源很难直接利用，也就造成了后续处理的复杂度增加。在转化 ID 的过程中，没能按计划仿照基因富集分析，直接找到水稻基因注释库，因此只能通过 NCBI 上的 gene_info 转换 ID。

对于共句显示的代码 sentence.py，第一次撰写时嵌套了多层循环，发现运行时十分缓慢。后来尝试了利用键值对来降低循环层次，却没能到达自己想要的输入效果，其速度也没有得到很大的提高（猜测主要为 re.search 本身造成的时间较长）。因此，采用了分割文献的方式，多进程处理，最终节约了 4/5 的时间。

对于实现相关可视化的代码，是在共句显示得到的文件基础上进行的。通过编写相关 perl 代码挖掘该文件里可能存在的生物学关系。代码虽然较之前的代码容易，但是在构思代码该实现什么样的功能上花了一点时间，且由于并不是经常编写代码，在代码实现的过程中会犯一点比较粗心的错误。

6.4 生物信息学实验设计的构思和体会

任何的实验设计，其终点都是通过这次实验设计能够得到什么样的实验结果。本次的生物信息学实验设计也是如此，在进行数据搜集的时候，为了佐证实验结果的可靠性，选择了 20000 篇关于水稻基因-性状的文献。为了实现已经构思好的可视化思路，编写代码得到共句的相关信息，并基于此选择合适的可视化方法。

最初打算使用课堂教授的方法来获取数据，但最终遇到了各种各样的问题。通过开辟新方法，了解了 edirect 等工具多样的处理方法。并且对如何处理大批量数据造成的过长时间有了更深的体会。

6.5 人员分工

陶芳婷：代码：数据可视化，文章撰写，使用 LaTeX 进行文章排版。
黄婷婷：代码：数据获取及处理，绘制网络图，文章撰写。

7 附录（如有）

S1. 2021 年课程网页. `https://hzaubionlp.com/course-bionlp-and-kd/`

S2. 课程论文可以使用的基础代码，可参考 github 页面, `https://github.com/huangtingting123/`

参考文献

[1] Wei Chih-Hsuan. Pubtator central: automated concept annotation for biomedical full text articles. *ucleic acids research*, 2019.

[2] Jonathan Kans. Phd.entrez direct: E-utilities on the unix command line. *NCBI*, 2013.

6

针对 Covid-19 的文献挖掘和知识发现

> 尝试一下从 *PubMed* 中挖掘实体吧。
>
> – Jingbo Xia

项目要求：

分析 PubMed 数据库提供的 Covid-19 文本摘要，围绕基因、突变、化合物等实体，进行实体抽取，对获得的实体进行知识挖掘和展示。

提示：使用 PubTator 获取相关实体。`https://www.ncbi.nlm.nih.gov/CBBresearch/Lu/Demo/tmTools/curl.html`

相关论文三篇：

吴恒《Covid-19 科学文献知识发现》

余思克《Covid-19 科学文献知识发现》

孙阳黄紫嫣《Covid-19 科学文献知识发现》

6.1 吴恒《Covid-19 科学文献知识发现》

新型冠状病毒肺炎 (Corona Virus Disease 2019，COVID-19)，简称“新冠肺炎”。2020 年 2 月 11 日，世界卫生组织总干事谭德塞在瑞士日内瓦宣布，将新型冠状病毒感染的肺炎命名为“COVID-19”。根据现有病例资料，新型冠状病毒肺炎以发热、干咳、乏力等为主要表现，少数患者伴有鼻塞、流涕、腹泻等上呼吸道和消化道症状。重症病例多在 1 周后出现呼吸困难，严重者快速进展为急性呼吸窘迫综合征、脓毒症休克、难以纠正的代谢性酸中毒和出凝血功能障碍及多器官功能衰竭等。值得注意的是重症、危重症患者病程中可为中低热，甚至无明显发热。轻型患者仅表现为低热、轻微乏力等，无肺炎表现。从目前收治的病例情况看，多数患者预后良好，少数患者病情危重。老年人和有慢性基础疾病者预后较差。儿童病例症状相对较轻。本课题通过对 pubtator 上关于 Covid-19 的 gene，chemical，mutation 三种实体进行提取，进而获取知识发现。

Covid-19科学文献知识发现

吴恒

华中农业大学信息学院，430070，武汉，湖北，中国

摘要

新型冠状病毒肺炎（Corona Virus Disease 2019，COVID-19），简称“新冠肺炎”。2020年2月11日，世界卫生组织总干事谭德塞在瑞士日内瓦宣布，将新型冠状病毒感染的肺炎命名为“COVID-19”。根据现有病例资料，新型冠状病毒肺炎以发热、干咳、乏力等为主要表现，少数患者伴有鼻塞、流涕、腹泻等上呼吸道和消化道症状。重症病例多在1周后出现呼吸困难，严重者快速进展为急性呼吸窘迫综合征、脓毒症休克、难以纠正的代谢性酸中毒和出凝血功能障碍及多器官功能衰竭等。值得注意的是重症、危重症患者病程中可为中低热，甚至无明显发热。轻型患者仅表现为低热、轻微乏力等，无肺炎表现。从目前收治的病例情况看，多数患者预后良好，少数患者病情危重。老年人和有慢性基础疾病者预后较差。儿童病例症状相对较轻[1]。本课题通过对pubtator上关于Covid-19的gene，chemical，mutation三种实体进行提取，进而获取知识发现。

关键词: Covid-19, pubtator，gene，chemical，mutation

1 课题概况

选修本课程可以强化个人的数据提取分析能力，而这两种能力在生信的学习过程中必不可少。Covid-19是目前的一个热点，根据世界卫生组织最新实时统计数据，截至北京时间5月23日22时45分，全球累计新冠肺炎确诊病例166346635例，累计死亡病例3449117例[2]。可以说新冠肺炎的产生和传播对全世界都造成了不可估量的损失，选取这个热点课题通过对已有的Covid-19相关的文献深入了解Covid-19的特性，获取知识从而更好的对Covid-19进行研究，在充分了解Covid-19后帮助早日解决这个世界性的热点问题。

2 数据

从NCBI上获取Covid-19相关文章的pmid，litcovid是收录Covid-19相关文献的集合，收录于litcovid中的文献与课题所需要的数据更为合适，因此获得litcovid中文献的pmid即可。根据pmid从pubtator的实体库中提取本次课题所需要的三种实体，即基因，突变，化合物。

3 研究方法

3.1 研究方法的算法背景，与其他方法的联系与区别

首先通过pmid一一定位获取pubtator实体库中对应的研究对象的实体，再对于获取的实体进行频率统计绘制图云。Wordcloud算法就是找到文本中占比重最大的字体，设置一个基础字体。其余的字体比例根据该字体来计算，在可绘制区域随机放入比重最大的字体，保存当前的绘制区域。检测屏幕区域中

有字体的位置，把对应的像素标识为true，再取下一个word，计算当前word的width和height，先在之前绘制区域内找是否有足够的空间容纳当前word的width和height，如果有则放入，如果没有则在当前绘制区域的上方或者下方或者左方或者右侧来放入。放入后更新当前的绘制区域并重复这一步。

3.2 研究方法中的核心思路

Wordcloud算法就是找到文本中占比重最大的字体，设置一个基础字体。其余的字体比例根据该字体来计算，在可绘制区域随机放入比重最大的字体，保存当前的绘制区域。检测屏幕区域中有字体的位置，把对应的像素标识为true，再取下一个word，计算当前word的width和height，先在之前绘制区域内找是否有足够的空间容纳当前word的width和height（，如果有则放入，如果没有则在当前绘制区域的上方或者下方或者左方或者右侧来放入。放入后更新当前的绘制区域并重复这一步。

3.3 本文的方法部分与课堂讲授内容的联系和区别/或补充

课堂上讲授的实体获取方法是获取相关文献的pmid后，使用脚本获取对应文献的摘要，再对摘要合集文本进行处理整理抽取实体再进行分析。本文通过pmid直接对pubtator上的实体库进行实体获取。而本文进行的后续分析方法则与课堂上讲授的一致。生成词云使用的Wordcloud算法。

4 算法实践和代码编写要求

4.1 任务描述

初步代码编写的任务是把从NCBI上下载litcovid文献的对应pmid和从pubtator上下载文献实体合集，根据pmid一一对应，只抽取litcovid收藏合集中文献的实体。然后对实体进行频率统计并生成词云。

4.2 实验设计

首先对实体进行获取，再对实体进行频率统计并生成词云，根据实体出现频率以及对应的实体类型进行分析思考，得出有关的生物学意义。对实体在数据库网站上进行检索了解更多的信息进行深层次的学习。

4.3 某些关键代码

```
gene_result = [],with open('data/gene2pubtatorcentral', 'r', encoding='utf-8')
 as f:
    for line in tqdm(f):\n",
        geneid = line.strip().split('\\t')[0]\n",
        if geneid in textid:
        gene_result.append(line.strip().split('\\t'))

mutation_20_x = [x[0] for x in mutation_20],
mutation_20_y = [x[1] for x in mutation_20]

 plt.figure(figsize=(8, 4.5)),
    plt.bar(mutation_20_x, mutation_20_y),
    plt.xticks(mutation_20_x, mutation_20_x, rotation=30),
    plt.show()\n"
```

5　主要的生物信息学实验和实验结论

选取突变实体的频率分析结果作图片展示，如图1，其中D614G出现频率最高，生成的词云如图2。由此可以初步判断，D614G基因是Covid-19发生突变改变病毒的重要基因，从NCBI上检索D614G基因，该基因与病毒表面糖蛋白的合成有关，这更进一步证实了该基因确实与Covid-19的突变大有关联。在病毒颗粒的外部发现了尖峰糖蛋白，并赋予冠状病毒类似冠状的外观。这种糖蛋白介导病毒颗粒的附着和进入宿主细胞。S蛋白是疫苗研制、抗体治疗和诊断性抗原试验的重要靶点[3]。对于D614G基因的研究非常重要，是控制冰防范Covid-19的突变进度的一个重要切入口。

基因实体中出现频率最高的是ACE2，通过NCBI的检索得知ACE2是人类血管紧张素转换酶2相关的基因，COVID-19对于人体的破坏很有可能与此基因已经该基因编码的血管紧张素转换酶2有关，关于COVID-19的治疗药物可以从对血管紧张素转换酶2的保护入手。而频率紧随其后的angiotensin converting（血管紧张素转化）和converting enzyme（转换酶）也都与之相关，一定程度上能证明前面的观点。血管紧张素转化酶（ACE）又称激肽酶Ⅱ或肽基-羧基肽酶。属血管内皮细胞膜结合酶，由肽的C端将氨基酸切为两段变换而来，可使肽链C端二肽残基水解。ACE广泛分布于人体各组织，以附睾、睾丸及肺的含量较丰富，其中肺毛细血管内皮细胞ACE活性最高。它附着于内皮细胞表面可被分解释放入血循环。ACE主要功能有两个：

1. 催化血管紧张素Ⅰ转化为血管紧张素Ⅱ；

2. 使缓激肽失活。血管紧张素转化酶因这两种功能而成为治疗高血压、心力衰竭、2型糖尿病和糖尿病肾病等疾病的理想靶点。ACE测定方法主要有比色法、酶偶联法等。

化合物实体频率最高的是oxygen和water，相对而言实体过于具有普适性应该不具有太大的价值，进而参看之后的实体为lopinavir ritonavir，通过检索得知这是一种适用于与其他抗反转录病毒药物的联合用药可以用于HIV的治疗，而COVID-19同HIV一样也是反转录病毒（RNA病毒），可以推测这种药物在对COVID-19的患者的临床治疗中会起到重要的作用。

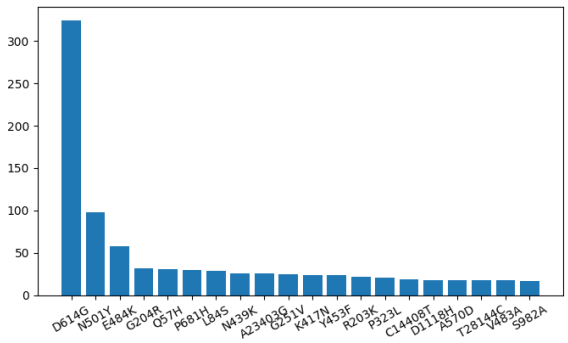


图1

图2

6 后记

6.1 课程论文构思和撰写过程

在撰写论文前首先对所需要的的数据进行获取，pubtator可以获取NCBI上关键词检索得到的文献的摘要，并且对实体有不同颜色的标注。在搜索Covid-19的相关文献时，发现pubtator网页上有一个litcovid的文献收藏合集，专门收录与Covid-19相关的文献，因此相较于直接检索Covid-19，选用litcovid中的文献收藏合集应该与Covid-19的相关性更强，所得到的分析结果也更准确更具有说服力。

最初是寄希望于直接从pubtator获取摘要合集再进行实体抽取。在pubtator上下载litcovid收藏的有关Covid-19的文献摘要合集后发现虽然下载的合集文件是XML格式但是实际打开后并非按照XML格式排版并且文献摘要合集中没有对于实体的特殊标注，这个方法就行不通。虽然文献摘要合集文件不符合我们对需要数据的要求，但是单个文献摘要XML文件的排版是合理的并且对于相关实体有标注，可以批量单个下载litcovid的文献摘要再进行实体抽取。这样虽然可行，但是由于需要对全部文献进行单个的操作，在有自知之明的情况下明白自己的代码能力有限，于是就也放弃了这个方案。

最后在浏览pubtator网页时发现网页有一个实体合集可供下载，并且对于每个实体都有对应的pmid一一标注，再结合上课时练习的pmid提取练习，解决数据获取的方案就呼之欲出了，直接从NCBI上下载litcovid文献的对应pmid，并从pubtator上下载文献实体合集，再根据pmid一一对应，只抽取litcovid收藏合集中文献的实体，两者都有一个总文件，因此操作起来方便且效率比方案二要高得多。

对实体的获取完成后就要进行分析，首先想到的当然就是频率统计，出现频率最高实体对于所要研究的对象来说必定是重中之重。首先经过频率统计的是突变实体，在获取的频率柱状图中，D614G一马当先，频率大幅度超过其他实体。在NCBI上检索D614G对该实体进行深度学习，D614G是与Covid-19外表糖蛋白有关的一个基因，病毒就是由外表蛋白和内部的遗传基因所构成，在病毒颗粒的外部发现了尖峰糖蛋白，并赋予冠状病毒类似冠状的外观。这种糖蛋白介导病毒颗粒的附着和进入宿主细胞。S蛋白是疫苗研制、抗体治疗和诊断性抗原试验的重要靶点[3]。由于是高频率的突变实体，D614G必然与Covid-19的突变息息相关。对于所有的高频率实体都可以通过数据库检索的方式获取相关信息并进行进一步的学习和分析。而所做的工作也可能出现一些无用功，比如化合物实体提取后统计数量中氧气和水的频率遥遥领先且远大于其他实体，但这两个实体太过于具有普适性。基本上所有的生物都需要氧气和水，因此这两个实体的统计一定程度上就不太有分析学习的意义。

6.2 所参考主要资源

获取实体部分的代码为原创，对实体做频率统计生产词云的代码参考于
https://amueller.github.io/word_cloud/
对实体的资料查询来自于NCBI数据库
https://www.ncbi.nlm.nih.gov/

6.3 代码撰写的构思和体会

本课题所需要的代码均有在课堂上讲述并在实验课练习过，并且上课时有讲述算法背后的原理。对于笔者这样代码能力较弱的同学非常友好，既能相对比较顺利的完成作业内容也可以从课程中学到原理知识，可以说是实践理论两丰收。

6.4 生物信息学实验设计的构思和体会

在获取所需要的的数据时，可以不必完全依靠爬虫脚本，最好先对数据所在的数据库网站多进行观

察了解，数据库平台会提供相应的下载功能，大部分所需要的数据应该是可以直接下载或者批量下载，比起脚本要更加方便更加有效率。在生物信息学的实验实践中，一个可行的想法或思路比掌握实际调用的算法工具更为重要，有了一个明确可行的方向才有机会去调用。通过对文献实体的分析，可以更加明确所需要研究的研究对象的重点，文本挖掘分析可以辅助帮助抓住研究的重点，提高工作效率。对于从文本挖掘中获得的重点可以反复进行检索学习，多层次的加深自身对其的了解，达到了解知识，深度学习的目的，不要仅限于浅尝辄止，不要吝啬于使用检索工具或搜索引擎，在信息开放的现在我们可以很容易的获取我们所需要的信息和知识，而重点挖掘就是引领我们进步的最快捷径。

参考资料

[1] 世卫组织：全球新冠肺炎确诊病例超过1.663亿例 *央视新闻*，2021

[2] 新冠肺炎与流感，该如何区分？ *人民网-人民健康网*，2020

[3] The spike glycoprotein is found on the outside of the virus particle and gives coronavirus viruses their crown-like appearance. This glycoprotein mediates attachment of the virus particle and entry into the host cell. S protein is an important target for vaccine development, antibody therapies and diagnostic antigen-based tests. *NCBI*, 2021

6.2 余思克《Covid-19 科学文献知识发现》

新型冠状病毒 (Covid-19) 是一种由严重急性呼吸道综合征冠状病毒 2 型引发的传染病，该病毒自 2019 年末出现以来，由于高传染性和高隐蔽性，很快在全球范围大规模爆发并急剧扩散，成为了人类历史上致死人数最多的流行病之一。疫情爆发一年多以来，有关 Covid-19 的研究成果在科研论文数量上已经具备了一定的规模。由于人工阅读文献获取最新研究成果的效率低下，针对探索 Covid-19 的入侵机理、了解基因突变与进化情况、研发治疗药物、研制疫苗等多种具有时效性的科学研究，使用新的技术快速获取论文中的研究成果信息这一需求极为迫切。因此，我们使用了生物自然语言处理领域的实体识别工具 PubTator，结合自己搭建的生物信息学分析流程，对 PubMed 收录的 122230 篇 Covid-19 相关研究文献进行文本数据挖掘，围绕基因、突变、化合物、物种、疾病等实体进行知识发现。通过研究，我们发现了 Covid-19 是一种严重的呼吸道传染病，趋向于侵染哺乳动物，有着致命性，其侵染通常伴随着感冒，咳嗽，与中风，急性肾损伤，糖尿病有着密切的联系。诸如“ACE2”，“spike”，“IL-6”等来源于人类或 Covid-19 基因组中的基因与 Covid-19 的侵染可能有着密切的联系。此外，我们还发现许多与 Covid19 相关的基因与化合物之间有着潜在的互作关联，并构建了关联网络。通过生物医学文本挖掘与知识发现的手段，我们基于大量文献数据得出了许多与 Covid19 有关的信息，这些信息对于后续疫苗的研发，病毒作用机理，药物研制等研究提供了一定的参考。

课程论文 GitHub 网址：`https://github.com/kiekie233/BioNLP-course/`

Covid-19科学文献知识发现

余思克[1]

[1]华中农业大学信息学院，430070，武汉，湖北，中国

摘要

新型冠状病毒（Covid-19）是一种由严重急性呼吸道综合征冠状病毒2型引发的传染病，该病毒自2019年末出现以来，由于高传染性和高隐蔽性，很快在全球范围大规模爆发并急剧扩散，成为了人类历史上致死人数最多的流行病之一[1]。疫情爆发一年多以来，有关Covid-19的研究成果在科研论文数量上已经具备了一定的规模。由于人工阅读文献获取最新研究成果的效率低下，针对探索Covid-19的入侵机理、了解基因突变与进化情况、研发治疗药物、研制疫苗等多种具有时效性的科学研究[2, 3]，使用新的技术快速获取论文中的研究成果信息这一需求极为迫切[4, 5]。因此，我们使用了生物自然语言处理领域的实体识别工具PubTator，结合自己搭建的生物信息学分析流程，对PubMed收录的122230篇Covid-19相关研究文献进行文本数据挖掘，围绕基因、突变、化合物、物种、疾病等实体进行知识发现。通过研究，我们发现了Covid-19是一种严重的呼吸道传染病，趋向于侵染哺乳动物，有着致命性，其侵染通常伴随着感冒，咳嗽，与中风，急性肾损伤，糖尿病有着密切的联系。诸如“ACE2”，“spike”，“IL-6”等来源于人类或Covid-19基因组中的基因与Covid-19的侵染可能有着密切的联系。此外，我们还发现许多与Covid19相关的基因与化合物之间有着潜在的互作关联，并构建了关联网络。通过生物医学文本挖掘与知识发现的手段，我们基于大量文献数据得出了许多与Covid19有关的信息，这些信息对于后续疫苗的研发，病毒作用机理，药物研制等研究提供了一定的参考。

关键词: Covid-19，生物医学，实体识别，文本挖掘与知识发现

1 课题概况

生物医学自然语言处理是一门利用NLP的自动化方法和手段解决生物信息领域的一些数据挖掘问题并进行知识发现的学科。由于自己在生物信息学本科阶段的学习中发现自己的兴趣更倾向于计算方面，希望以后从事人工智能在生物医学中的应用等研究，因此选修了该课程，选课的目的是为了学习NLP理论与技术，学习相关的机器学习与深度学习模型，开拓视野，为将来的科研与工作打下基础。

新型冠状病毒（Covid-19）疫情自2019年爆发以来，对各国产生了严重的影响，且仍未得到有效控制。已发表的相关研究文献有数十万篇，人工阅读文献归纳研究成果是低效且不现实的，因此，我选择了本课题，目的是使用生物医学自然语言处理的PubTator工具，结合自己搭建的分析流程，高效挖掘文献摘要中的实体信息并进行知识发现，为Covid-19的研究提供一定的参考。本项目计划挖掘出与Covid-19相关的高频基因、化合物、人类基因组和Covid-19基因组的突变情况、Covid-19侵染的物种以及Covid-19感染造成的症状；了解Covid-19相关高频基因的功能以及Covid-19相关突变对Covid-19侵染造成的影响；了解基因与化合物的关联并构建关联网络。

2 数据

本项目的实验数据来自于PubTator，该工具整合了PubMed文献数据库中收录的文献的摘要中包含的实体信息[6]。我们通过edirect工具获取了PubMed文献数据库中所有与“Covid-19”关键词有关的文献的uid，并通过调用PubTator的API接口，根据文献的uid，下载了这些文献的实体识别信息。获取的文献摘要实体

数据由三个部分组成，第一个部分是文献的标题，第二个部分是文献的摘要，第三个部分是文献中识别出来的实体。实体内容包括基因、化合物、DNA变异、蛋白质变异、SNP、物种、疾病七个种类。实体数据来源于122230篇Covid-19相关文献的摘要。本项目的研究将重点围绕实体数据中的摘要部分以及实体部分进行展开。

3 研究方法

3.1 研究方法的算法背景，与其他方法的联系与区别

本项目的实体识别过程使用的是PubTator工具。PubTator是一款基于Web，可通过使用高级文本挖掘技术来加快人工文献的管理（例如，注释生物实体及其关系）的工具[7]，作为一个多合一的系统，其提供了一站式服务来注释PubMed引用[8]。与其他方法相比，PubTator提供了已经完成识别的实体数据，不需要自己完成搭建框架、训练模型、调参、然后进行实体识别等一系列过程。虽然PubTator针对实体识别的效果相较于BERT+CRF等神经网络训练出来的模型来说不占优势，但是其提供的一站式多合一服务，极大地简化了大家进行实体识别任务的操作过程，大大缩短了时间，适合快速展开研究[9]。

本项目的数据处理部分使用到了Linux Shell，R语言、Python语言以及相关的拓展包，相较于其他语言，Linux Shell可以方便地利用正则表达式快速进行数据的初步筛选，R语言以及Python语言具有强大的拓展包生态、绘图能力、数据处理能力以及简洁的语法，使得数据处理过程方便且迅速。

3.2 研究方法中的核心思路

本项目首先通过调用PubTator的API，获取了122230篇Covid-19相关文献的实体数据，并围绕基因、化合物、DNA变异、蛋白质变异、SNP、物种、疾病实体进行词频的统计，目的是分析哪些实体词汇在文献中得到了大量的报道，从而了解与Covid-19具有潜在关联的各类实体信息。然后，我们对高频基因等实体展开了进一步的探索，通过数据库了解基因功能，从而推测它们与Covid-19的入侵机制、作用机理等特征。

此外，我们还关注了基因与化合物之间的间接联系，通过对122230篇Covid-19相关文献的摘要数据，对基因与化合物实体进行共句分析，分析思路是：当某一个基因与一个化合物同时出现在了一句话中，即认为它们之间具有潜在的关联。通过编写脚本分析基因与化合物的关联并绘制关联网络，直观反映Covid-19相关的基因与化合物之间的互作关联（图 *1*）。

3.3 本文的方法部分与课堂讲授内容的联系和区别与补充

本文的方法部分与课堂相同点在于都使用了PubTator工具进行实体数据的获取。不同点在于实体数据的获取细节。课堂中通过文献搜索结果，手动复制几篇感兴趣的文献对应的uid，然后通过调用API下载实体数据，而由于本项目计划对所有的Covid-19相关文献进行实体数据的获取，一共有122230篇文献对应122230个实体数据集合需要被下载，很明显不能继续通过手动搜索并复制粘贴uid进行实体数据的下载了，于是本项目使用了edirect工具首先根据关键词自动化提取了所有Covid-19相关文献的uid，然后遍历读取uid列表文件调用PubTator API进行实体数据的下载，获得了大量丰富的实体数据。

除了课堂讲授的PubTator实体抽取基础分析，我们补充进行了各实体词频统计、基因功能分析、基因与化合物工具关联及其关联网络绘制等分析方法辅助于Covid-19文献的知识挖掘（图 *1*）。

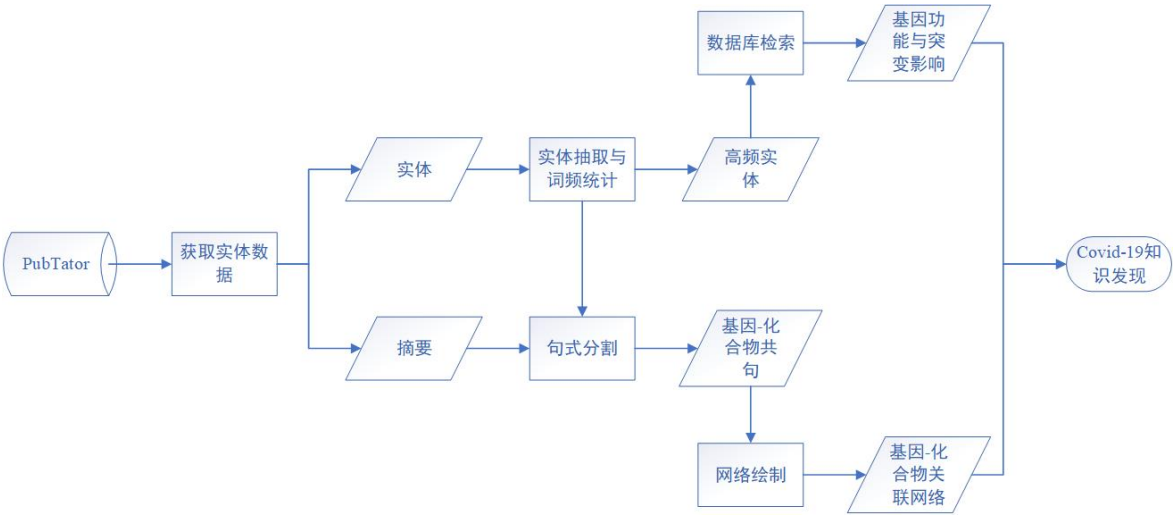


图 1: 项目流程图。图片来源：自绘，工具：Visio

4 算法实践和代码编写要求

4.1 任务描述

第一步通过编写Shell脚本进行数据的获取。此步骤将搜索与Covid-19相关的122230篇文献的uid，并通过遍历这些uid依次从PubTator中获取实体信息。由于相关的工具是Linux环境下的，所以使用Shell脚本能很好地完成此任务。

第二步通过编写Shell脚本进行数据的预处理。此步骤将通过Shell中的正则表达式把实体数据分为两个子数据，一个是实体数据，另一个是摘要数据，供后续数据分析使用。由于正则表达式是Linux Shell中强大的字符处理工具，故使用Shell编写脚本可以很好地完成此任务。

第三步通过R语言编写脚本读取实体数据，按照不同的实体种类把实体数据拆分开，统计实体词汇的词频并排序，然后根据词频分别绘制不同实体的词云。由于R语言强大的绘图能力以及统计能力，其非常适用于该统计分析及绘图任务。

第四步通过Python语言编写脚本读取上一步R语言处理过后的基因实体以及化合物实体数据，同时读取摘要数据，通过判断基因与化合物是否同时出现在一句话，确定基因与化合物的潜在关联，并格式化输出关联文件，后续将依据此文件绘制关联网络。由于Python丰富的内置函数以及其拓展包Pandas，其能够很好地完成遍历、判断算法以进行共句分析及格式化输出。

4.2 实验设计

数据的获取是通过Shell中的edirect工具中的esearch命令完成，它将获取所有与Covid-19相关文献的uid。然后使用Shell中的curl命令，调用PubTator的API接口，依次下载这些文件。数据的预处理是通过使用正则表达式结合grep与sed命令，分别进行实体与摘要的筛选，得到待分析的数据。

实体数据的分析是通过R语言中的数据框结合table，sort等函数进行词频统计，然后通过wordcloud2函数进行词云的绘制。共句关联分析是通过Python中的pandas包结合基本函数实现。

代码在运行之前需要安装相关的依赖包，R语言需要安装wordcloud2包，Python需要安装Pandas包，且需要关注文件地址信息，根据自己的数据存放地址调整代码中的读写地址。本项目使用的R为4.0.3版本，Python为3.9.5版本。

4.3 关键代码

```r
#代码来源：https://github.com/kiekie233/BioNLP-course/blob/main/script/Get_entity_freq.R
#词云绘制（以基因实体词频大于一百的词汇为例）
gene=freq[freq$bioconcep=="Gene"&freq$Freq>=100,]#筛选高频词汇
order_temp=order(gene$Freq,decreasing = T)
gene=gene[order_temp,]#排序
rownames(gene)=seq(1,nrow(gene))
gene$Freq=gene$Freq^0.4
wordcloud2(gene,size=0.6)#调整参数绘制词云
```

```r
#代码来源：https://github.com/kiekie233/BioNLP-course/blob/main/script/Get_entity_relation.py
#共句判断算法（以每一句话为单位，判断基因与化合物在摘要中的共句信息）
for gene in gene_list[0]:
    for chemical in chemical_list[0]:
        print("Gene:",gene,"chemical:",chemical)
        for sentence in sentences:
            if gene in sentence and chemical in sentence:
                relation=relation.append({"gene":gene,"relation":"relate","chemical":chemical
                    },ignore_index=True)#计算基因与化合物的共句信
                    息
```

5 主要的生物信息学实验和实验结论

5.1 实体词频统计

通过对各实体词频的统计，我们发现物种实体显著富集于“人类”，“猪”，“老鼠”，“猫”，“哺乳动物”，“鸡”等词汇上，可以从中发现这些物种可能是实验验证或者是病例观察得出的Covid-19主要侵染的物种。可以推测，Covid-19偏向于侵染哺乳动物。疾病实体富集于“侵染”，“呼吸道”，“致命”，“死亡”，“糖尿病”，“感冒”，“疼痛”，“气喘”，“急性肾损伤”，“中风”，“神经”等词汇上，可以发现Covid-19是一种传染性疾病，其具有致命性，其感染伴随着感冒，疼痛，气喘等症状，可能对肾脏，神经系统等组织与器官造成伤害，且糖尿病与其关联密切。基因实体显著富集于“ACE2”，“spike”，“IL-6”等词汇上，可以推测这些基因与Covid-19有着密切的联系，可能是Covid-19病毒的组成结构基因，也可能是辅助其侵染宿主的关键功能基因。突变、化合物等实体词汇的聚集同样反映出了许多与Covid-19有关的信息。在此，我们展示出基因的词频可视化信息，完整的结果可在本项目的GitHub仓库中查看并获取。（图 2）

图 2: 基因词频词云。图片来源：自绘，工具：R

5.2 基因功能分析

通过对基因实体的词频统计，我们得到了一组频率较高的基因，针对这批基因，我们通过NCBI数据库对其进行了检索，得到了这些基因的具体信息。（表*1*）

这些基因中一部分来源于人类基因组，一部分来源于Covid-19病毒基因组，来源于人类基因组的基因所编码的蛋白，很可能作为媒介协助Covid-19进行侵染，或者是作为Covid-19的攻击目标靶点；而来源于病毒基因组的基因，很可能是病毒的结构基因，其编码的蛋白具有一些特征可供识别，也可能是功能基因，辅助病毒入侵宿主。后续可以针对这些基因进行深入研究，为疫苗的研发，病毒作用机理，药物研制提供一定的参考。

表 1: 部分高频基因功能信息

基因	功能	物种
ACE2	血管紧张素转换酶2	人类
spike(S)	刺突糖蛋白	Covid-19
IL-6	白介素6	人类
CRP	C反应蛋白	人类
TMPRSS2	跨膜丝氨酸蛋白酶2	人类
Mpro	ORF1a多蛋白	Covid-19
CD4	CD4抗原	人类
CD8	CD8抗原	人类
N	核衣壳蛋白	Covid-19

5.3 基因与化合物互作分析

通过对基因与化合物在摘要中的共句分析，我们筛选出了共句次数超过100次的基因-化合物对，我们认为共句次数超过100次足以证明该基因与化合物之间存在着一定的潜在关联。

其中，“S-CO”，“N-CO”，“ACE2-CO”，“S-iron”，“S-oxygen”，“S-water”，“IL-6-CO”等基因-化合物共句频率非常高，可以推测它们之间有着较大的可能性存在互作关联关系，后续可以针对这些高频率的基因-化合物关联对进行进一步深入的研究以探究Covid-19的特征。（表*2*）

表 2: 部分高频基因-化合物共句信息

基因	化合物	频率
S	CO	81014
N	CO	48717
ACE2	CO	1289
S	iron	1544
S	oxygen	1194
S	chloroquine	905
S	water	737
IL-6	CO	726
S	hydroxychloroquine	652

这些化合物可能与基因表达的产物会有互作，可能会产生激活、抑制、失活作用。这些关联具有潜在的研究意义，可以为药物靶点筛选，Covid-19病毒入侵机制的研究提供参考。对此，我们绘制了基因-化合物关联网络，直观展示它们的关联信息。（图 *3*）

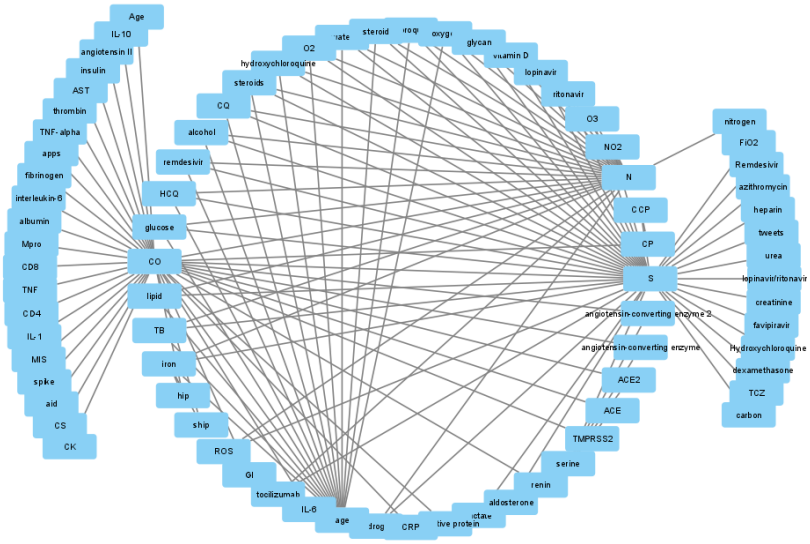


图 3: Covid-19相关基因-化合物关联网络。图片来源：自绘，工具：Cytoscape

6 后记

6.1 课程论文构思和撰写过程

在进行课程论文的构思之前，我首先深度思考了一下该项目的细节信息：这个项目是什么？目的是什么？为了达到目的，需要做些什么？如何展示自己的结果？经过了一段时间的思考，我认为这个项目是针对海量的Covid-19论文进行实体抽取，发现其中潜在的，隐含的知识。此项目的目的是为了解决人工阅读文献了解信息费时费力且效率不高这一难题，并基于大量数据挖掘Covid-19的信息，为致病机理，入侵方式，药物靶点，疫苗研发等科学研究提供一定的辅助与参考。为了达到目的，需要尽可能将所有与Covid-19有关的文献全部下载下来，然后进行数据分析，最后把结果充分生动地展示出来。于是基于这些思考，我对论文中所需要呈现的内容以及呈现方法有了大致的思路：将大量的文献摘要、实体信息通过整理、可视化，直观地反映出来，并辅以精简的文字描述，做到言简意赅，图文并茂。

论文的撰写过程还是比较痛苦的，因为我需要做的不仅是把自己的结果照搬上来，我需要把项目的前因后果，实现方式，以及对于结果的讨论和自己观点的提出，通俗易懂且清晰地表达出来，这对我的归纳总结能力以及发散性思维是一个重大的考验。所以在撰写的过程中，我不断地打草稿，修改，修改，再修改，最终尽自己最大努力完成了这篇论文的撰写。

6.2 所参考的主要资源

完整项目储存在个人GitHub中：`https://github.com/kiekie233/BioNLP-course/`。项目代码均为自己编写。GitHub仓库中包含了该项目的思路、所需数据、完整代码、项目的完整结果等信息。

本项目的生物信息学流程、完整的代码及其运行方式、依赖的环境、所需要的包、运行顺序、实现的功能、可能存在的报错等帮助信息均在GitHub中有详细介绍，可以供大家更好地进行本项目分析流程的复现，欢迎大家复现本项目并提出问题、探讨见解、批评指正。

结果文件中除了上文中展示的数据之外，还包含着因篇幅限制而未能展示在文章中的数据。额外的结果包含：所有种类实体的词频信息及其可视化词云数据、基因-化合物共句信息、基因-化合物共句频率信

息等。Github中提供的完整数据可以供大家做额外的、自己感兴趣的拓展分析。

非常感谢老师和师兄师姐的课堂教学以及提供的课程资源，感谢同学们针对于项目思路以及技术实践方面的探讨与帮助，虽然本项目由我个人完成，但老师、师兄师姐以及同学们潜移默化中给了我许多帮助，这门课让我受益颇丰，再次表示感谢。

6.3 代码撰写的构思和体会

在生物信息本科阶段的学习中，我发觉自己对于代码编写有着强烈的热爱。每当需要实现一个功能，在编写代码之前，需要先把这个功能拆分为几个板块，然后分板块实现这些小目标，然后把它们拼凑在一起。例如共句分析的代码构思，我将其分为了四步，第一步是对基因、化合物进行遍历，第二步是对摘要中拆分好的每一句话进行遍历，第三步是将基因、化合与与摘要中的每一句话进行匹配，第四步是结果的输出。

虽然在代码的构思过程中，头脑中思维逻辑的飞快转动使我头痛不已，但是当自己完成了debug，实现了自己预想的功能，顺利跑通代码之后，内心有着无法用言语形容的一种畅快感。尤其在自己的代码绘制出了精美的图片，巧妙完成了各类计算，解决了生物学问题时，我领悟到了代码的魅力，交叉学科的魅力。代码虐我千百遍，我待代码如初恋，不论在编写代码时，遇到了多大的困难，只要逻辑清晰，稳扎稳打，冷静内心，都能写出漂亮的代码（当然也得保持不错的代码风格，方便维护以及共享）。我想，今后我依旧会保持对代码的热爱，因为它真的太神奇了。

6.4 生物信息学实验设计的构思和体会

因为自己在去年疫情期间尝试过针对Covid-19的序列进行一定的分析，以及自己有过一定的生信项目经历，所以自己对Covid-19及其实验设计有着一定的了解。本项目的关键目的是替代人工阅读，从文献摘要中挖掘出与Covid-19有关的知识。所以，在得到实体数据之后，我们首先关注的是实体的频率并根据频率发现一些规律与特征，进而尝试探索不同实体之间存在的潜在关联，深入挖掘实体相关的信息。同时，还要充分地利用得到的觉果，将数据以图片的形式生动地展现出来。

设计生物信息学实验最重要的不是写代码，而是好好地把握住生物问题，首先需要弄懂我们为什么要研究、开展这个项目，这个项目的目的是弄清楚什么科学问题。只有把生物学问题把握住了，才能更好地设计生物信息学实验，将生物学问题转化为概率问题，计算问题，算法问题等，从而对其进行实践，这便是我理解的交叉学科的一个特点：跨学科的科学问题转化以及多种类的技术工具应用。

7 附录

S1. 本项目的GitHub页面. `https://github.com/kiekie233/BioNLP-course`

S2. 2021年BioNLP课程网页. `https://hzaubionlp.com/course-bionlp-and-kd/`

S3. PubTator网页. `https://www.ncbi.nlm.nih.gov/research/pubtator/`

参考文献

[1] Thirumalaisamy P Velavan and Christian G Meyer. The covid-19 epidemic. *Tropical medicine & international health*, 25(3):278, 2020.

[2] T Thanh Le, Zacharias Andreadakis, Arun Kumar, R Gómez Román, Stig Tollefsen, Melanie Saville, Stephen Mayhew, et al. The covid-19 vaccine development landscape. *Nat Rev Drug Discov*, 19(5):305–306, 2020.

[3] Xuetao Cao. Covid-19: immunopathology and its implications for therapy. *Nature reviews immunology*, 20(5):269–270, 2020.

[4] Gemelli Against COVID, Post-Acute Care Study Group, et al. Post-covid-19 global health strategies: the need for an interdisciplinary approach. *Aging Clinical and Experimental Research*, page 1, 2020.

[5] Thinh Hung Truong, Mai Hoang Dao, and Dat Quoc Nguyen. Covid-19 named entity recognition for vietnamese. *arXiv preprint arXiv:2104.03879*, 2021.

[6] Chih-Hsuan Wei, Robert Leaman, and Zhiyong Lu. Beyond accuracy: creating interoperable and scalable text-mining web services. *Bioinformatics*, 32(12):1907–1910, 2016.

[7] Chih-Hsuan Wei, Hung-Yu Kao, and Zhiyong Lu. Pubtator: a web-based text mining tool for assisting biocuration. *Nucleic acids research*, 41(W1):W518–W522, 2013.

[8] Chih-Hsuan Wei, Alexis Allot, Robert Leaman, and Zhiyong Lu. Pubtator central: automated concept annotation for biomedical full text articles. *Nucleic acids research*, 47(W1):W587–W593, 2019.

[9] Donald C Comeau, Chih-Hsuan Wei, Rezarta Islamaj Doğan, and Zhiyong Lu. Pmc text mining subset in bioc: about three million full-text articles and growing. *Bioinformatics*, 35(18):3533–3535, 2019.

6.3 孙阳黄紫嫣《Covid-19 科学文献知识发现》

Covid-19 是近两年以来最具影响力的话题，目前已有许多相关报道及文献，通过抓取相关文献能得到 Covid- 19 中极具研究价值的实体信息。在本次项目中，通过 PubTator 对 123596 篇文献中的实体进行识别、提取，对所得实体进行知识挖掘，从而推测出 Covid-19 的产生与基因、疾病、突变、化合物实体之间的关系，分析 Covid-19 的产生原因及其影响，并给出 Covid-19 预防及治疗的建议。

课程论文 GitHub 网址：`https://github.com/HZYShadow/BioNLP-course-Covid-19`

Covid-19 科学文献知识发现

孙阳 [1], 黄紫嫣 [2]

[1] 华中农业大学信息学院，430070，武汉，湖北，中国
[2] 华中农业大学信息学院，430070，武汉，湖北，中国

摘要

Covid-19 是近两年以来最具影响力的话题，目前已有许多相关报道及文献，通过抓取相关文献能得到 Covid-19 中极具研究价值的实体信息。在本次项目中，通过 PubTator 对 123596 篇文献中的实体进行识别、提取，对所得实体进行知识挖掘，从而推测出 Covid-19 的产生与基因、疾病、突变、化合物实体之间的关系，分析 Covid-19 的产生原因及其影响，并给出 Covid-19 预防及治疗的建议。

关键词: Covid-19, PubTator, 实体, 相关性

1 课题概况

自然语言处理是计算机科学领域与人工智能领域中的一个重要方向，它主要研究实现人与计算机之间通过自然语言进行有效通信的各种理论和方法。选修自然语言处理是为了学习到平时接触较少的新知识，开阔眼界，获得新技能，以此合理应用在自己在做的项目及之后将进行的研究之中，优化研究方法，通过文本挖掘，更高效地获得知识发现，避免浪费大量时间做可以通过自然语言处理高效完成的工作。

Covid-19 的出现引起了人们的广泛讨论，目前已有许多学者投身于对其的研究之中，相关文献数目也非常之多。尽管报道很多，但我们对 COVID-19 的认知还是十分浅显的，选该题目是希望能更深入了解 Covid-19，从文献中挖掘到有趣的知识。本文主要阐述从 PubTator 中识别相关文献的实体并对其进行提取，主要选取基因实体进行分析，并通过对化合物、疾病和突变的进一步研究，找出这些实体与 Covid-19 的联系，得出 Covid-19 及其突变最可能与哪些实体相关的结论，利用这些关系对 Covid-19 的产生做出合理推断，并对 Covid-19 的预防及治疗提出建议。

2 数据

通过 NCBI 的 edirect 工具中 esearch 命令抓取 PubMed 中 Covid-19 相关文献的 PMID，用 xtract 工具将其转换成纯文本格式的内容存入文档中，此时共抓取到 123596 个 PMID。该数据获得时间为 4 月 20 日，但截止至 4 月 20 日 PubMed 上共有 125459 篇相关文献，与抓取到的 PMID 数目存在差距，但不知其中的原因。

编写 shell 脚本 NCBI.sh 通过 PubTator 抓取与已获得的 PMID 对应的文献信息，并将其保存至文本文档中。抓取到的文献摘要数量与 PMID 数目相同，为 123596 篇，同时抓取结果中还包括标题以及 PubTator 识别出的基因、化合物、突变等实体。

3 研究方法

3.1 研究方法的算法背景，与其他方法的联系与区别

PubTator 是一个基于网络的生物医学文本挖掘工具，与 Brat、NLPIR 汉语分词系统等实体识别工具相类似，都能用于实体标注与实体识别。这些实体识别工具一般采取基于词典的方法、基于统计机器学习的方法或基于词典结合机器学习的方法进行实体识别及标注。而与其他实体识别工具不同的是，PubTator 更符合文本挖掘经验有限的生物管理者的需求，它提供了在计算机辅助生物固化中生成自动计算机预注释的最新功能，同时它适用于不同的注释任务，还允许其用户个性化自己的注释环境。PubTator 中，GeneTUKit 用于基因提取；GenNorm 可对基因进行标准化；DNorm 、SR4GN 、tmVar 分别用于注释疾病、物种、突变；基于字典的查找方法可用于识别化合物 [1]。而通过 Linux 的正则表达式操作，可以对 PubTator 识别及标注的内容进行过滤，筛选出不同的实体信息。

R 是一套完整的数据处理、计算和制图软件系统，可用于数据存储和处理系统，数组运算，统计分析及制图，操纵数据的输入和输出，实现分支、循环，同时用户也可自定义功能。使用 R 能对提取出的实体信息进行统计分析，绘制图表，而通过对行与列的处理以及循环操作，能找出特定信息，以此进行知识挖掘。

3.2 研究方法中的核心思路

对 Covid-19 相关文献 PMID 进行提取。利用 edirect 工具获取 PubMed 中与 Covid-19 相关文献的 PMID，输出得到 PMID 的文本文件。

根据得到的 PMID 抓取对应文献摘要及实体信息。借鉴老师课堂讲授的 shell 脚本对得到的 PMID 通过 PubTator 抓取文献，利用 PubTator 我们能得到 PMID 对应的文章标题、摘要。同时 PubTator 能对实体进行识别及标注，从而我们能得到文献中对应的实体信息，其中包括基因、疾病、突变、化合物等 6 类实体的信息。

抽取出实体，对不同实体进一步进行知识挖掘。利用正则表达式过滤文章标题及摘要，并分别提取出不同类型的实体存入以实体类型名称命名的文本中。分别对基因、疾病、突变、化合物这 4 类实体中的数据出现频率进行统计，得到每类实体中出现次数最多的实体，其中重点对基因实体信息进行研究，找出其在染色体上的分布情况，并对其进行 GO 富集分析。我们找出了前 13 个与 Covid-19 显著相关的基因，并找出了与这些基因出自同一篇文献中的疾病和突变，以此探究这些基因与 Covid-19 哪类突变相关，Covid-19 可能与何种疾病同时出现，同时通过对化合物的分析，找出有哪些化合物能对 Covid-19 起到治疗作用。

3.3 本文的方法部分与课堂讲授内容的联系和区别/或补充

通过 edirect 工具中的 esearch 抓取 PMID 这一方法是从课堂上老师对其的介绍而得知，但由此方法得到的 PMID 文本存在一定问题，该问题是通过查找相关资料而解决的；实现通过 PubTator 获取文献摘要和实体信息的 shell 脚本来自于课堂老师介绍的脚本，但这里我们简化了其中部分代码；而对实体进行知识挖掘的代码由我们使用 R 语言编写而成。

4 算法实践和代码编写要求

4.1 任务描述

目前在 PubMed 上直接下载获取 PMID 的上限为 10000，远小于与 Covid-19 相关文献数量，故通过 edirect 工具中的 esearch 抓取 PMID。在得到 PMID 后，利用 PubTator 得到对应文献摘要及实体标注，并通过正则表达式分别提取出六类实体信息，以便进一步对不同实体进行挖掘。因为 R 在数据处理、计算

和制图上功能较为强大，且能操纵数据的输入和输出，实现分支、循环，所以接下来我们使用 R 语言统计各实体出现次数并绘制出柱状图，对频次由高到低进行排序，提取出出现次数较高的实体信息，以便进一步进行实体知识挖掘。统计出基因实体在染色体上的分布情况，并对基因进行 GO 富集以此研究其功能及作用。在 R 中用 for 循环找出与 Covid-19 显著相关的基因出现在同一篇文献中的疾病和突变且统计其出现频率，以此推断疾病和突变体对 Covid-19 产生的影响。根据化合物统计分析的结果，在 NCBI 中查找相关信息，并结合基因、疾病以及突变的特点，找出能用于治疗 Covid-19 的化合物。

4.2 实验设计

第一步，是获取 PMID。使用 edirect 工具中的 esearch 命令获取与 Covid-19 相关的 PMID，并将其存入 txt 格式的文件中。因 windows 和 Linux 系统下文件之间会存在一些差别，不能直接将 esearch 命令提取到的 PMID 在 PubTator 上进行文本挖掘，需要使用 dos2unix 进行转换，此时得到的 PMID 文件才能用于实体识别。

第二步，是编写 shell 脚本通过 PubTator 获取 PMID 对应文献摘要及识别得到的实体信息。在课堂上老师教授的 shell 脚本的基础上通过 PubTator 进行文本挖掘，再使用正则表达式过滤所得文件中的标题、摘要等部分，分别提取出六类实体信息存入以其实体名称命名的文本中。

第三步，是对获得实体进行知识挖掘。统计基因实体出现频率并使用 ggplot() 命令作出柱状图，找出其在染色体上的分布情况，对其进行 GO 富集分析。结合获取的文献摘要找出与 Covid-19 感染及预后诊断显著相关的 13 个基因，以 PMID 为中间桥梁，使用 R 语言中的 for 循环找出与这些基因出现在同一篇文献中的疾病、突变实体，统计这些实体出现频率，以此推断哪些疾病与突变和 Covid-19 息息相关。统计化合物实体出现频率，找出出现次数较多的化合物，通过文献摘要及 NCBI 相关信息进一步了解这些化合物的功能，从而对 Covid-19 的预防及治疗提出建议。

4.3 某些关键代码

```
代码来源: https://github.com/HZYShadow/BioNLP-course-Covid-19
# 获取PMID
esearch -db pubmed -query "covid-19" | efetch -format uid > pubmed_pmid.txt
代码来源: https://github.com/HZYShadow/BioNLP-course-Covid-19
# 提取出实体，过滤掉文章标题和摘要部分
grep -vE "[0-9]{8}\|[ta]\|" result_2.txt > entity.txt
grep -E "\bGene\b" entity.txt  > gene.txt
代码来源: https://github.com/HZYShadow/BioNLP-course-Covid-19/blob/main/bionlp.R
# 画出现500次以上基因实体的频率直方图
gene_table<-as.data.frame(table(entitylist[["gene"]]$ID))
p<-ggplot(subset(gene_table,Freq>500),aes(Var1,Freq))
p<-p+geom_bar(stat = "identity")
p<-p+theme(axis.text = element_text(angle = 45,hjust = 1))
代码来源: https://github.com/HZYShadow/BioNLP-course-Covid-19/blob/main/bionlp_find.R
# 找出与显著基因位于同意篇文献中的疾病，其中disease是提取出的疾病实体，diff_gene是显著基因
tmp <- c()
for( i in diff_gene){tmp <- append(tmp,which(gene$V6 == i)) }
tmp2 <- gene[tmp,]
pmid <- unique(tmp2$V1)
tmp3 <- c()
for (j in pmid)  { tmp3 <- append(tmp3,which(disease$V1 == pmid)) }
final_disease <- disease[tmp3,]
```

5 主要的生物信息学实验和实验结论

使用 esearch 命令，我们获取到共 123596 个 PMID，与 PubMed 上搜索到与 Covid-19 相关结果数量 125459 有一定差距，虽不清楚其中的具体原因，但我们推测可能与文本的可用性以及文章类型相关，在抓取过程中因其造成部分文献的丢失。编写 shell 脚本以此通过 PubTator 挖掘 PMID 对应的文献信息以及 PubTator 识别出的实体，利用正则表达式过滤掉文献标题和摘要部分，提取出其中的实体部分，并根据实体类型分别提取出六类实体信息。我们聚焦于基因实体，通过对获取的基因实体的 ENTREZID 进行处理、筛查，最终获得 2854 个基因。接着，在 NCBI 上对 ENTRZEID 进行批量查询，最终获得基因的详细信息并存为文本文件，该文件中包括物种来源、基因 ID、状态、名称及别名、染色体位置等基因信息，并对每一个基因进行了一定的描述。我们对这些基因在染色体上的分布情况进行了统计，并作出对应的基因分布图。

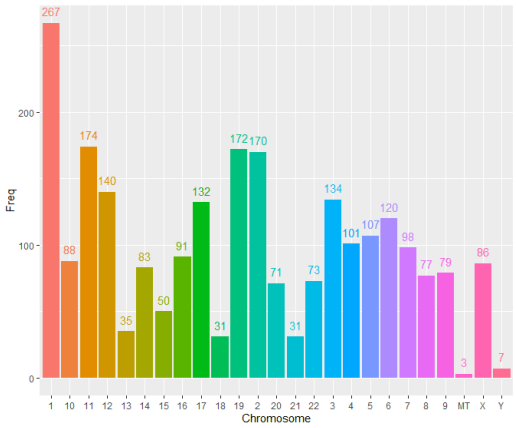


图 1: 基因在染色体上的分布情况。图片来源：自绘，工具：Rstudio 。

如图1所示，其中 1 号染色体上的基因实体数目远多于其他染色体，共有 267 个基因在 1 号染色体上；2、11、19 号染色体也较多，其数量都在 170 个左右；而 Y 染色体和线粒体染色体上基因实体非常少，几乎不存在有对应的基因实体。通过这一数据结果，我们可以初步推断 Covid-19 的产生极大可能与 1、2、11、19 号染色体相关，而与 Y 染色体和线粒体染色体的关系不大，但是如果在之后的研究中发现造成 Covid-19 的基因来自于 Y 染色体和线粒体染色体，则致病基因可以直接锁定在两个染色体上这 10 个基因之中。为了进一步探究与 Covid-19 相关的基因的功能和作用，我们对其进行了 GO 富集分析，得到结果如图2所示。从富集结果可看到基因主要富集于对细菌的反应、细胞活化调节等生物学过程，以及受体调节活性、信号转导受体激活剂等分子功能相关，并且绝大部分基因位于膜上以及囊泡腔中，这也说明 Covid-19 极大可能靶向到膜结构，直接与膜上受体作用或参与到细胞运输过程中，影响这些分子功能、生物学过程，使体内的调节失常，以此表现出患病特征。而 KEGG 富集结果表明这些基因主要和细胞因子与细胞因子受体的相互作用、EB 病毒感染等相关，这一结果也能支持我们做出的猜想。

接着，统计基因出现频次，并绘制出现 500 次以上的基因频次直方图，从图3中我们可以直观看到出现次数较多的基因。我们提取出其中出现次数较多的基因，并结合摘要信息，手动筛选出 13 个与 Covid-19 感染及预后诊断显著相关的基因。这 13 个基因中 ACE2 是最为显著的基因，它在各种人体器官中表达，其编码的蛋白是人类冠状病毒 HCoV-NL63 和人类严重急性呼吸系统综合症冠状病毒 SARS-CoV 和 SARS-CoV-2 的突触糖蛋白的功能性受体，与 Covid-19 存在有很大的联系。另外，已显示其编码的蛋白质是能够在患有自身免疫性疾病或感染的人中引起发烧的内源性热原。而 IL6 与 COVID-19 的预后诊断有关，该基因编码在炎症和 B 细胞成熟中起作用的细胞因子，其功能涉及多种与炎症相关的疾病状态。筛选出的这些基因的功能都直接影响 Covid-19，我们认为这些基因与 Covid-19 的产生密切相关。为了进一步研究这些基因与 Covid-19 的关系，我们通过找出与这些显著基因位于同一篇文献中的疾病和突变，以此推断出突变是

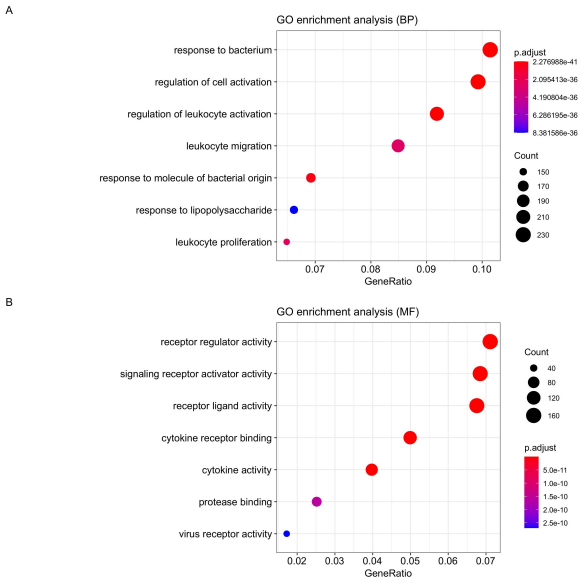


图 2: GO 富集分析结果。图片来源：自绘，工具：Rstudio 。

否与这些基因相关，同时推测 Covid-19 可能与哪些疾病并发出现。通过 R 语言实现该操作，我们发现并不存在与显著基因共文献的突变，这仅仅只说明突变与这些显著基因关联度较小，而可能存在其他基因与其作用从而引发 Covid-19，只是我们暂时未挖掘出结果。我们在 NCBI 中查找了抓取到的文献中提及较多的突变，这些突变主要为新冠病毒的刺突 S 蛋白上的突变，其中 D614G 突变是 614 这一位点上蛋白发生突变，由 D 变为了 G，且该突变已被证实具有更强的感染力。

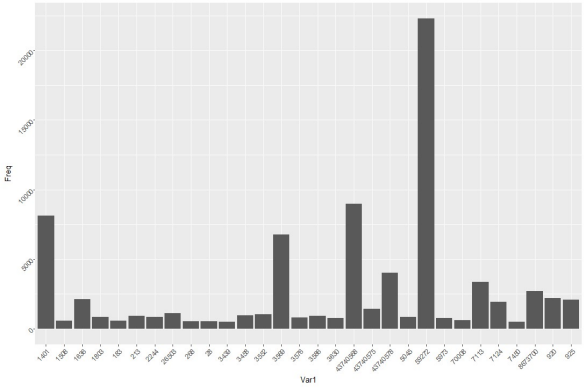


图 3: 基因频次直方图。图片来源: 自绘，工具：Rstudio。

而在找与显著基因共文献的疾病时可以发现，出现频次最高的疾病是 Covid-19，这也证明了这些显著基因与 Covid-19 的产生密不可分。从表1中我们可以看到，慢性阻塞性肺疾病、白细胞增多、淋巴细胞减少、MDS 以及严重急性呼吸系统综合征冠状病毒 2 型感染也与显著基因出现在同一篇文献中，这些症状很有可能是 Covid-19 的并发症，Covid-19 极有可能是由这些疾病发展而产生的，也可能 Covid-19 导致了这些疾病，这些疾病或许能成为检测 Covid-19 的信号，而在治疗 Covid-19 的同时也要更加注意以此避免引发这些疾病。为了推测出对 Covid-19 较为有效的治疗方法，我们对化合物进行分析，表2为出现次数前 10 的化合物。出现次数较多的化合物大体上可分为两类，一类是 ACE2 等基因的靶标化合物，而另一类是抗 RNA 病毒感染的化合物。我们知道 Covid-19 是一种 RNA 病毒，且与 ACE2 等基因相关，这证明这些化合物很大程度上能对 Covid-19 起到作用，能将其中一些用于 Covid-19 的治疗。例如，羟氯喹的 TLR7 通过识别微生物特有的分子模式来控制宿主对病原体的免疫反应，可以用于治疗 Covid-19。而 Remdesivir 是

一种新型抗病毒核苷酸类似物，可以用于治疗 RNA 病毒感染，这对研究 Covid-19 的治疗方法具有极大价值。

表 1: 与显著基因出现在同一篇文献中的疾病及其出现次数

Disease	Frequency
chronic obstructive pulmonary disease	2602
COVID-19	15612
leukocytosis	2602
Lymphocytopenia	2602
MDS	2602
Severe Acute Respiratory Syndrome Coronavirus-2 Infection	2602

表 2: 出现次数前 10 的化合物信息

Name	Frequency	description
hydroxychloroquine	15465	抑制血浆血红素聚合酶，靶标为 DNA, 正在研究用于治疗 SARS-CoV-2
oxygen	9142	-
Remdesivir	6684	受体为 Replicase polyprotein 1ab, 用于治疗 RNA 病毒感染
tocilizumab	6211	用于治疗发炎和自身免疫性疾病, 目前正在研究以治疗 COVID-19 的重症患者
chloroquine	5753	为原型抗疟药，但机理尚不清楚，正在研究用于治疗 SARS-CoV-2
Vitamin D	3999	具有预防或治疗动物疾病的常见作用
alcohol	3814	-
water	3631	-
Azithromycin	3464	一种半合成大环内酯类, 靶标为 23S 核糖体 RNA
Lopinavir	3219	是一种抗逆转录病毒蛋白酶抑制剂，靶标为人类免疫缺陷病毒 1 型蛋白酶

以上这些结果对 covid-19 预防及监控的研究来说，是具有一定参考价值的，可以针对这些染色体或基因进行监控，将其作为判断是否有 Covid-19 的一项指标，同时，或许可以通过这些化合物找到 Covid-19 的新治疗方法。

6 后记

6.1 课程论文构思和撰写过程

首先为了得到 PMID，并通过 PubTator 抓取文献，我们使用了老师上课教授的 shell 脚本，但用 esearch 命令获取的 PMID 无法成功抓取到对应文献，我们发现这是由于得到的 PMID 文件出现了不符合要求的换行符造成的，这时需要使用 dos2unix 对 PMID 文件进行转换后才能进行实体识别。老师讲授的 shell 脚本抓取文献的时间较长，对约 12 万文献的提取至少需要花几天的时间来完成，所以我们希望能找到其他方法来缩短这一时间。我们通过编写 Python 并行程序进行文本挖掘，效率的确大幅度提高，但是也存在一个问题，shell 脚本中用 8s 是为了更安全地从网站上挖掘文本，这里使用并行程序相当于多线程同时抓取，从本质来说和将 8s 缩短的区别不大，其安全性有待考量，故未采取该方法。我们尝试了 R 中 pubmed.miner 进行实体识别，理论上来说这种方法能较快的到文献对应的实体，但无论如何尝试 pubmed.miner 包中函数的使用都出现了报错，无法进行实体识别操作。我们又通过 R 中 pubtatordb 包从 pubtator 数据库中直接获取实体信息，但使用该方法也出现了错误，多次尝试后得到的实体仍为空，且出现了非常多的报错信息。最终，我们仍采用 shell 脚本进行文本的抓取。

在成功提取出实体后，我们首先对每一类实体都做了出现次数统计，得到了每类实体中出现次数最多的实体。我们聚焦于与 Covid-19 直接相关的基因实体，对基因进行了 GO 富集分析。我们绘制出现次数在 500 以上的基因的柱状图，找出出现次数较多的基因，根据文献及注释内容，我们手动挑选出 13 个与

Covid-19 感染及预后诊断关系最为密切的基因，认为这些基因是 Covid-19 相关的显著基因。我们以这些显著基因为标准，分别找出与这些显著基因出现在同一篇文献中的疾病、突变实体，这里我们并未发现任何突变与这些显著基因相关，但我们不认为这代表着突变对 Covid-19 的影响不大，这仅仅只代表这些显著基因可能与突变不存在联系。这也说明我们挖掘到的信息存在有一定的局限性，需要进一步改进。在找出这些实体与实体、实体与显著基因之间的关系后，我们想通过 Cytoscape 画出关系网络图，但出现较多一个节点对应几千甚至更多关系，绘制出的关系网络非常杂乱，美观性和实用性都较差，因此我们未展示绘制出的关系网络图。我们也尝试通过 Bert 实现 Covid-19 文献的词汇嵌入和展示，对代码进行调试后顺利跑通，但未设计好进一步研究方案，故未对其继续进行研究。

根据所挖掘到的信息，我们认为 13 个显著基因与 Covid-19 的产生及影响有直接或间接的关系，这些基因很有可能位于 1、2、11、19 号染色体上，但可以较为肯定地认为不会存在于 Y 染色体和线粒体染色体上。在研究中可以着重研究这些染色体及基因，以此找到新的控制 Covid-19 的方法。同时 Covid-19 与一些疾病相关，这些疾病可能是 Covid-19 的并发症，也可能因为这些疾病会引起 Covid-19，通过对这些疾病的监测和治疗，或许能起到预防 Covid-19 的作用，而这些疾病或许能作为判断是否可能患 Covid-19 的一项指标。对化合物进行分析中，我们能看到羟氯喹、伦地西韦等化合物与 Covid-19 关系较密切，或许能从这些化合物出发，研制出靶向定位或抗 RNA 病毒感染的药物用于 Covid-19 的治疗。

6.2 所参考主要资源

1.PubTator 相关介绍：`https://academic.oup.com/nar/article/41/W1/W518/1105731`

2. 查找实体相关资料：`https://www.ncbi.nlm.nih.gov/`

3.GO 富集分析参考代码链接：`https://github.com/bionlp-hzau/Tutorial_4_GO_Enrichment`

4.pubtatordb 使用参考：`https://github.com/MAMC-DCI/pubtatordb/`

5. 课程项目代码主要使用 R 语言自己编写而成，已上传至 Github：`https://github.com/HZYShadow/BioNLP-course-Covid-19`

6.3 生物信息学实验设计的构思和体会

一开始时，我们花了较长时间用于缩短文献抓取的时间，但发现尝试的三种方法可行性都不高，最后选择使用 shell 脚本提取文献。对于整体实验的设计一开始是比较迷茫的，我们对所有实体都统计出出现频次，但在夏老师的启发下我们逐渐找到了研究方向，开始聚焦于基因实体，以其为出发点进行深入研究，进而找出显著基因，并找出与显著基因同时存在于同一篇文献中的疾病、突变，研究这两者与基因的关系，从而推测出这些实体与 Covid-19 的关系，并大胆提出自己对 Covid-19 的见解。在实验中我们使用到了许多从课堂中学到的知识，也通过自己的实际操作加深了对自然语言处理的理解，从中受益良多。

6.4 人员分工

孙阳：负责主要代码编写（PMID 及文献抓取、Python 并行程序编写、绘制各实体频次柱状图、绘制基因在染色体上的分布情况、基因 GO 富集分析、找显著基因）。

黄紫嫣：负责课程论文撰写，部分代码编写（R 实现文献抓取、找出与显著基因存在于同一篇文献中的疾病及突变）。

参考文献

[1] Pubtator: a web-based text mining tool for assisting biocuration. *Nucleic acids research*, 2013.

7

针对老年痴呆症的文献挖掘和知识发现

换一个疾病试试 *PubMed* 实体抽取和知识挖掘吧？

– Jingbo Xia

项目要求：

分析 PubMed 数据库提供的 AD 文本摘要，围绕基因、突变、化合物等核心词汇的语法和依存路径，结合所挖掘的生物实体，进行知识挖掘和展示。

提示：使用 PubTator 获取相关实体。`https://www.ncbi.nlm.nih.gov/CBBresearch/Lu/Demo/tmTools/curl.html`

使用依存数信息，参考项目链接：`https://github.com/bionlp-hzau/tutorial4dependencytree`

相关论文一篇：

张富卿李佳桐《老年痴呆症科学文献的核心词汇，语法和依存路径分析》

7.1 张富卿李佳桐《老年痴呆症科学文献的核心词汇，语法和依存路径分析》

本实验的主要目的是探究有关阿兹海默症的有关性质，初步以“疾病”、“基因”、“突变”、“药物”、“细胞系”五个方面来对其进行探究。在此过程中，我们逐渐将目标关注在“药物”和“基因”之间，通过建立最短依存路径以及互作网络图的方法来显示各词的密切程度和关联，并且以特定药物 (Donepezil(多奈哌齐)、Rivastigmine (卡巴拉丁)、Memantine Hydrochloride(盐酸美金刚) 为靶点，查询其具体功能，将之前的结果细化，进一步得到特点药物与多种基因的关系。

课程论文 GitHub 网址：`https://github.com/xichutong/NLP-AD.git`

老年痴呆症科学文献的核心词汇、语法和依存路径分析

张富卿 [1], 李佳桐 [2]

[1] 华中农业大学信息学院，430070，武汉，湖北，中国
[2] 华中农业大学信息学院，430070，武汉，湖北，中国

摘要

本实验的主要目的是探究有关阿兹海默症的有关性质，初步以“疾病”、“基因”、“突变”、“药物”、“细胞系”五个方面来对其进行探究。在此过程中，我们逐渐将目标关注在“药物”和“基因”之间，通过建立最短依存路径以及互作网络图的方法来显示各词的密切程度和关联，并且以特定药物（Donepezil（多奈哌齐）、Rivastigmine（卡巴拉丁）、Memantine Hydrochloride（盐酸美金刚））为靶点，查询其具体功能，将之前的结果细化，进一步得到特点药物与多种基因的关系。

关键词：阿兹海默症，最短依存关系，多奈哌齐，卡巴拉丁，盐酸美金刚

1 课题概况

华中农业大学的选课系统其实一直比较模糊，又因为本专业的信息在网上很难找到，所以选课的时候只能通过课程名、任课老师、学分三项进行选择。之所以选这门课其一是因为学分合适，其二就是因为是夏静波老师任教，毕竟之前也上过老师的课，对老师的讲课方式和形式比较熟悉。之所以选当前的课程论文题目，其实也是因为在刚开课的时候，对于所提供的课程论文题目的具体内容也是一头雾水，不知道它们究竟要如何操作、如何实验，所以所有不太熟悉的内容中选择一个相对看得懂的题目。

就本篇文章来说，我们认为只能算是照猫画虎，是对于课程内容的熟悉、学习、练习，通过实践的方式加强对于所学知识的理解。只敢说是亦步亦趋地遵循课程内容进行实验。也是表现我们对于本次分析流程的浅显理解，从某种角度上来说也是自曝其短了。

2 数据

使用 Elasticsearch 的 esearch 功能从 Pubmed 数据库中检索有关阿兹海默症疾病的相关文章

Elasticsearch 是一种分布式可扩展的、建立在全文搜索引擎 Apache Lucene(TM) 基础上的实时搜索和分析引擎，主要有四种功能：

1. 全文搜索
2. 分布式实时文件储存，并将每一个字段都编入索引，以便其可以被搜索
3. 实时分析的分布式搜索引擎
4. 可以扩展到上百台服务器，处理 PB 级别的结构化或非结构化数据

Elasticsearch 使用倒排索引，相比于关系型数据库的 B-Tree 索引，其查找速度更快。

传统的 B-Tree 为了提高查询的效率，减少磁盘寻道次数，将多个值作为一个数组通过连续区间存放，一次寻道读取多个数据，同时也降低树的高度，即【Term Dictionary】。

Elasticsearch 为了能快速找到某个 term，将所有的 term 排个序，二分法查找 term，logN 的查找效率。类似于 B-Tree 通过减少磁盘寻道次数来提高查询性能，Elasticsearch 直接通过内存查找 term，不读磁盘，但如果 term 太多，庞大的【Term dictionary】也不适于放于内存中，于是采用【Term Index】，类似字典里的索引页。【Term index】不需要存下所有 term，而只是他们的一些前缀与【Term Dictionary】的 block 之间的映射关系，再结合 FST(Finite State Transducers) 的压缩技术，可以使 term index 缓存到内存中。从 term index 查到对应的 term dictionary 的 block 位置之后，再去磁盘上找 term，大大减少了磁盘随机读的次数。FST 以字节的方式存储所有的 term，这种压缩方式可以有效的缩减存储空间，使得 term index 足以放进内存，但这种方式也会导致查找时需要更多的 CPU 资源。Elasticsearch 要求 posting list 是有序的，这样做的一个好处是方便压缩，原理就是通过增量，将原来的大数变成小数仅存储增量值，再按照 bit 排序，最后通过字节存储。

其索引思路就是将硬盘里的数据尽量转移到内存中，减少磁盘随机读取次数，同时利用磁盘顺序读取特性，最大可能的利用内存。

在使用 Elasticsearch 进行索引时有 3 点注意事项：

1. 不需要索引的字段，要明确定义出来，因为默认是自动索引的。
2. 对于 String 类型的字段，不需要 analysis 的也需要明确定义出来，因为默认也是会 analysis 的。
3. 选择有规律的 ID 很重要，随机性太大的 ID(比如 java 的 UUID) 不利于查询。

3 研究方法

3.1 研究方法的算法背景，与其他方法的联系与区别

在本次实验中，我们主要采取了 pubtator 摘取文章，spacy 进行信息的比对、查找关联，Cytoscape 进行网络图绘制，进行最短依存树分析以及最短路径分析。

3.2 研究方法中的核心思路

我们希望根据有限的几个关键词性从所摘取的文献中获取相应生物信息的短句，通过主谓宾位置的分类构建依存树以及寻找最短路径，从而获得最简洁、直接、强烈的关系脉络，从而找出相关性最强的成分，进而获得有关阿兹海默症的治疗、发病、生成原因的情况。

3.3 本文的方法部分与课堂讲授内容的联系和区别/或补充

我们使用的方法的核心即课上讲解的关于依存树和最短依存路径的方法，结合老师课上讲解的文本摘取和依存树作用原理，将它应用在构建未知的阿兹海默症的“容貌”的实验中。

4 算法实践和代码编写要求

4.1 任务描述

本次实验代码的主要要求就是“摘取信息”，我们使用通过 shell 命令摘取文章和重要信息语句，通过最短依存树的路径分析寻找关联性最强的信息。

4.2 实验设计

我们使用的工具主要有两种，Linux Shell 系统和 Python，最初的文本摘取和定位词汇是在 linux 系统上进行操作，而后续的最短依存树路径分析则是以 Python 语言写成。

5 主要的生物信息学实验和实验结论

5.1 下载文章

下载 Elasticsearch version7.12.1:https://github.com/topics/elasticsearch (Elasticsearch 运行需要 java 环境)。解压安装在普通用户之下（elasticsearch 可以接受用户输入的脚本并且执行，出于系统安全的考虑，其无法在 root 用户下运行，否则会出现报错）.

在输入命令的过程中，我们试图对下载文章的范围进行限制，网站上的有关文章有 16 万余篇，最初的想法是将所选取的文章的年份限制在 2021 年内，但下载后发现年份限制命令没有起到作用，仍为 16 万篇，所以准备在后续步骤中进行限制。

```
1
2 代码来源: esearch说明书
3
4 esearch -db pubmed -query "Alzheimer's disease" | efetch -format medline efilter -mindate 2020 >Pm.txt
5 #未完全起效果的命令
```

5.2 提取文章 PMID

添加命令，将所提取的 PMID 限制在 10000 篇以内。

```
1
2 在原课上所给予的NCBI.sh代码上进行的改进和条件限制
3
4 i=$[i+1]
5         if [ $i -gt 10000 ];then
6             break
7         #读取10000行
```

5.3 使用 pubtator，根据 PMID 从响应文章中抽取词汇

PubTator 是一种基于网络的工具，用于加速手动管理文献。通过使用先进的文本挖掘技术来注释生物实体及其关系)。作为一个一体机的系统，PubTator 为注释 PubMed 引用提供一站式服务。

我们计划从文本库中以“cellline”、“chemical”、“disease”、“gene”、“mutation” 5 个目标范畴为搜索目的，摘取相应词汇以及其在文本中的位置。

首先，为了研究阿兹海默症的相应信息，在可能的情况下，我们认为“细胞系”反应了该病的发病位置，也许其病症集中作用在某一个细胞中，也许是多个细胞的集群相互作用；“化学药物”可能反应了与该病有关的药物治疗，关键的化学元素和化学反应；而之所以在以阿兹海默症为研究目的的实验中仍以“疾病”为关键词，是想确定与阿兹海默症有关的并发症，或是否在阿兹海默症真正发生前，会有一些前置的病症出现；而对于“基因”的分析可能包含着该病的发病位置，和突变的可能原因；“突变”方面，表现阿兹海默症的可能形成原因，是否是某种正常细胞或基因变异而导致，或是其发生是否会引起变异。由此，我们选择这五项，从而了解阿兹海默症的相关信息。通过对 pubtator 实体的生物名词的统计，我们发现从数量

上来说，发现绝大部分文献都是与基因联系最为密切和频繁，与其相关的细胞器仅有十余个，而有关突变仅有七个。具体的筛选信息可见 github 相关文档。

从筛选情况来看：其中出现频率最高的几个词语均与神经系统有着重要的关系。如 tau 蛋白，名为微管蛋白，是神经细胞骨架成分，而 tau 的异常过度磷酸化正是 AD 发病的主要原因。又如 ApoE（载脂蛋白），是一种多态性蛋白，参与脂蛋白的转化与代谢过程，而 ApoE 多态性与 AD 有着密切的联系，与淀粉样前体蛋白（神经毒性蛋白）的亲和力很高。

然而，当我们筛选之后，我们发现在我们获得正确信息的同时，“不速之客”也不请而来。我们发现该网站本身所做的关于成分的分类是较为模糊的，所以在筛选结果中有很多意料不到的成分混入。

例如在该系统中，它会把 protein（蛋白质）识别为 gene 的一种，或者是把 H20（水分子）归类为化学药物的一种（当然水确实属于化学元素的一种，但可以说和治疗阿兹海默症没有太大关系），从此也可以反映这个摘取的算法和网站的分类还是比较模糊的，无法完全按照需要的要求准确的达到目的，需要在后续分析中进行筛选。

所得部分信息：

表 1: Gene

Gene	number
Abeta	4579
tau	3303
APOE	906
PS1	704
Tau	682
amyloid-beta	55

表 2: Celline

Celline	number
SH-SY5Y	301
BV2	107
PC12	106
BV-2	67
HT22	37
N2a	31

表 3: Chem

Chem	number
lipid	449
water	354
glucose	345
cholesterol	344
iron	329
calcium	311

表 4: Mutation

Mutation	number
R47H	39
P301S	37
P301L	30
E280A	23
A673T	19
C677T	12

5.4 从筛序句子中获取关键词信息

在统计出所有关键词后，我们需要确定相关句子中出现这个词汇并非孤立偶然的，所以我们进一步筛选至少同时具有两个相关词汇的句子，假如这个句子中只有一个相关词汇，那么这个词的出现可能仅仅是偶然的，而两个同时出现的情况下，一方面二者之间可能具有关联，例如新的突变可能导致新的并发症的产生，不同细胞位置的突变影响可能不同，不同基因的突变影响可能不同；不同的药物可能对于阿兹海默症的治疗效果和作用机理不同，不同的药物可能通过作用于不同的细胞系内、通过不同的路径过程从而对阿兹海默症产生治疗效果。当然，这都是我们的猜想，但在最后，也许这些猜想有的可以证明，有的难以实现。

在这一步骤中，我们遍历所有句子，并试图从主宾位置中寻找联系，例如我们以药物为主体，查找其与其他几个部分的可能联系，同时这几种词的所在位置也必须位于主谓宾的结构位置上，这样才能确保所筛选的句子是着重描述这几项的有用语句。在这里我们使用 python 语言的 spacy 包来进行分析，spacy 是世界上最快的工业级自然语言处理工具，支持多种自然语言处理基本功能，主要功能包括分词、词性标注、

词干化、命名实体识别、名词短语提取等等。最终我们从 9 万余句子中筛选出 1064 条符合条件的成对关键词以及有关语句。

5.5 绘制网络图

根据上一步中我们筛选出的成对关键词，我们试图以关系网络图的形式来展示它们之间的复杂关系，这里我们使用了 Cytoscape 来完成关系图的绘制。这里我们截取了图的一部分，可以看出，多数元素在整个系统中都是相互关联存在的，孤立存在的元素不多，由此可见，我们可以大致将图中的元素单词基本认定为阿茨海默相关生物名词的一个系统关系。从个体来看，存在某些节点在整个系统中非常忙碌，有着密集的度关系，可见这个名词必然在阿兹海默症中扮演着重要角色。

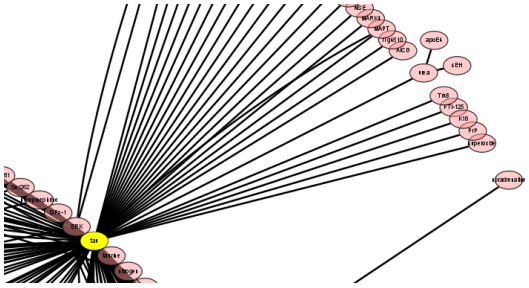


图 1: 自绘

这里我们截取了图的一部分，可以看出，某些元素是在整个系统中有非常忙碌的、复杂的关系，这也代表着它在本次实验中，在阿兹海默症中扮演着重要得角色。

5.6 对成对关系进行依存路径分析

所谓依存关系，一个中心词与其从属词之间的非对称关系，修饰词为从属词，被修饰词为支配词。将一个句子中所有词的依存关系以有向边的形式表现出来，即是依存句法树。依存关系树具有唯一、连通、无环、投射的特性。

短语生成依存树的过程是，从叶节点开始，把表示具体单词的结点归结到表示词类的结点上；自底而上，把主词归结到父节点上；把全局的中心名词归结到根节点上。

实验初期，我们希望通过择出 root 结点，即全句中的关键谓语，与关键的主语词、宾语词联系，从而分辨两者的从属关系，但当进行到实验后期我们发现很多文章中的句子结构比较复杂，而该算法的择取方式较为机械，所筛出的 root 结点和叶结点虽然在结构上的联系较为紧密，但在实际的语义中，二者可能不具有相互产生影响的关系。例如，有一些摘取的语句为“研究证明，……”，而这个算法所摘取的 root 结点就是为“证明”，这时因为在语法结构上，有主从句之分，一个句子真正的关键信息可能会放在从句之中，而这是该算法力所不能及的。

在这里我们也进行了实际检测，随即选择了十条句子进行 root 节点的选择，而结果明显不尽人意，能够正确连接关键词的 root 节点仅有 2 个。这样的结果也反映了 root 节点并不能反应出实际关键词的关系。

所以，在这种情况下，我们只能退而求其次，使用 python 中的 networkx 包进行分析。net 包虽然不能准确地识别句子中的语法结构，但它的范围是被限制在我们所寻找几个所选词汇中，其通过最短路径的联系，可以找到单词中相互最紧密的路径，这也往往是句子中所要描述表达的关键部分。从而可能获取到更为相关的、有用的信息。

而最终，通过最短路径筛选出的句子信息，我们发现仅有当两个关键词为并列关系时，最短路径反应的信息不多，但是从整体而言，有用信息的覆盖度依旧有很大意义（结果可见 github 相关文档）。

5.7 检索固定药物的相关信息

在这一步中，我们试图根据筛选出来的有关信息的句子和匹配出来的相应信息的关联之间，寻找有关药物的相关信息。例如，“memantine:AChE memantine shown exerted inhibition of ache”，通过这句话，我们可以的得知美金刚胺对疼痛有抑制作用，这可能被用于对病症得缓解。

从得上一步中所得到的各关键词间的 1064 条最短路径关系中，我们选取了三个在目前对阿兹海默症治疗最有效的药物。Donepezil（多奈哌齐）、Rivastigmine（卡巴拉丁）、Memantine Hydrochloride（盐酸美金刚）。以这三者为靶点筛选出了 27 条其与基因的关系。选取其中其中一条，” memantine shown exerted inhibition of ache”，其中大致信息为美金刚有着对 AchE 的强烈抑制作用，由此我们可以清楚美金刚与 AchE 两种化合物之间的具体关系。具体的 27 条信息大致如上述语句，可见 github 文档。

1. 多奈哌齐（Donepezil），本品属六氢吡啶类氧化物，是第二代特异的可逆性中枢乙酰胆碱酯酶（AChE）抑制剂，对外周 AChE 作用很小。本品通过抑制 AChE 活性，使突触间隙乙酰胆碱（ACh）的分解减慢，从而提高 ACh 的含量，改善阿尔茨海默病（AD）患者的认知功能。抑制乙酰胆碱酯酶活性的强度是抑制丁酰胆碱酯酶的 570 倍，具有较高的选择性。口服 10mg/kg 可对脑内胆碱酯酶产生抑制作用，且呈剂量效应关系。

2. 卡巴拉汀（Rivastigmine），通过抑制乙酰胆碱酯酶来增加脑中释放胆碱能的神经元的功能，从而改善阿尔茨海默病患者的认知作用。在脑内的海马和皮质区有高度选择性作用。此外，本品还可以减慢淀粉样蛋白 p—淀粉样前体蛋白（APP）片段的形成。而淀粉样斑块是阿尔茨海默病的主要病理特征之一。本品作用强度中等，比毒扁豆碱作用弱。与心脏、骨髓肌相比，对脑中乙酰胆碱酯酶具有更特异性。

3. 盐酸美金刚是由德国 Merz 公司研制的痴呆症治疗药物，是一种新型、低中度亲和力、电压依赖、非竞争性 N - 甲基 - D - 天冬氨酸（NMDA）受体拮抗药，可非竞争性阻滞 NMDA 受体，降低谷氨酸引起的 NMDA 受体过度兴奋，防止细胞凋亡，改善记忆，是新一代改善认知功能的药物。

6 后记

6.1 课程论文构思和撰写过程

就本次实验来说，我们的目的是寻找有关阿兹海默症的相关信息，在真正操作之前，一切都是未知的，我们对整个计划只有一个大概的轮廓，不知道过程中会发现什么，最后究竟能得出什么结论。根据要求，我们选择“突变”、“细胞”、“疾病”、“细胞”、“基因”五个方面来展示阿兹海默症的部分面貌，像是盲人摸象，我们通过不同的不同来摸索对象所能展现的形象。

但是整个过程并非顺利，一方面是我们对所用方法的不熟悉，例如；

另一方面，是对于所得的数据的疑惑，如上文所述，pubmed 网站对于文章中的词性的分类是粗糙机械的，这导致我们所摘取的词汇中有大量的非目的词汇。而在进行依存路径分析的时候，从某种角度上来讲，结果是成功的，我们找到了中心词，找到了与其相关的从属词，找到了它们之间的最短路径，理论上我们可以进行关联性的分析，但另一方面，我们也发现，结构上最有关联的、连通性最强的词之间，也许不一定有含义上的紧密关联，我们以“疾病”和“药物”为靶点，在语句中选取其作主词和宾语的部分，我们所希望的目的是找到“某药物能治疗某病”的语句，但经过筛选后发现，其中大部分是

而在绘制出的网络图中，我们也发现，各个词组的联系其实较为分散，并未如想像般的联系紧密，从而能使我们找出中心和重点。

6.2 所参考主要资源

参考的主要资源是《生物文本挖掘与知识发现概论》以及 CSDN 网站上对于有关算法解释和理解

6.3 代码撰写的构思和体会

整体代码实现之后，我们感觉难度不高，主要使用的是正则表达式与 spacy 的内容，但是实践过程中遇到的细节问题较多。首先是数据上有多层，从 pubtator 的整体数据到摘要的整体数据，再到每篇文献的每段句子，最后到每段句子中的每个词，数据量不断筛选，因此我们代码上采取了分步的操纵，一次一次的筛选可用信息，将每个阶段的信息都做了完整的存储，能够达到很好的回溯效果。

其次，在匹配搜索上，信息较为复杂。随着正则表达式不断深入，对于匹配搜索信息，越来越熟悉，对于正则表达式也更为熟练。

然后是对于字典的使用，为了联系更多的信息，在关系存储中，我们使用了复合的字典形式，也就是以字典的形式存储于字典中，最终以两个键对应一个值的结构，这个的构思来源于 perl 中的哈希的哈希，相关的还有哈希的数组与数组的哈希，虽然 python 中并未明确定义，但是这样的结果是允许存在的，通过这个结构，我们将数据更为清晰的联系在了一起。

本实验所涉及代码详见： https://github.com/xichutong/NLP-AD.git

6.4 人员分工（如有）

李佳桐：代码编写
张富卿：论文撰写

参考文献

[1]《Elasticsearch - 基础介绍及索引原理分析》. 神一样的存在. https://www.cnblogs.com/dreamroute/p/8484457.htm

[2] 盐酸美金刚（词条）. https://baike.baidu.com/item/

[3] 卡巴拉汀（词条）. https://baike.baidu.com/item/

[4] 多奈哌齐（词条）. https://baike.baidu.com/item/

[5]《生物文本挖掘与知识发现概论》. 夏静波.

8

针对 AGAC 语料库的序列标注方法探讨

AGAC 是我们课题组自己开发的一个专属语料库，试一试我们提供的序列标注工具吧。

– Jingbo Xia

项目要求：

使用序列标注工具或者神经网络代码，完成 AGAC 数据库的序列标注任务。并利用该任务所获得的模型，针对新文本进行挖掘。

提示：需编写脚本对 AGAC Track 的数据进行预处理，完成 Task 1.

参考论文链接：`https://www.aclweb.org/anthology/D19-5710/`

参考的 Wapiti 项目链接：`https://github.com/bionlp-hzau/Tutorial_4_CRF`

参考的 BiLSTM+CTF 项目链接：`https://github.com/bionlp-hzau/LSTM_CRF_useAGAC`

参考的 BERT 项目链接:`https://github.com/bionlp-hzau/BERT-CRF-for-BioNLP-OST2019-AGAC-Task1`

相关论文三篇

吴思琪苏馨如《对 AGAC 数据库的序列标注任务与新文本挖掘》

江毅炜《针对 AGAC 数据库的序列标注》

胡昕昀、沈敖《基于 BiLSTM 和 CRF 的序列标注》

8.1 吴思琪苏馨如《对 AGAC 数据库的序列标注任务与新文本挖掘》

AGAC 做为主攻基因突变与疾病之间的关系的语料库，目前已按照不同级别的语言信息、语义信息完成对相关文本的标注，是良好的参考与实验材料。CRF 做为神经网络出现前最好用的标注方法，拥有成熟的标注机制。在本实验中，我们将 CRF 分别与神经网络模型 LSTM 和 BERT 相结合，形成 LSTM-CRF 和 BERT-CRF 方法，将 AGAC 中的数据文件分为测试集和训练集，使用训练集分别对上述模型进行训练，并用测试集对训练得到的模型模型的标注效果分别进行评估。我们将 LSTM-CRF 和 BERT-CRF 方法训练模型的标注效果与 CRF 标注的效果进行对比，发现 LSTM-CRF 的标注结果与 CRF 差不多，BERT-CRF 方法训练模型的标注效果则要显著高于他们。最后我们使用 1045 篇宫颈癌文献的标题与摘要做为预测文本，代入 LSTM-CRF 和 CRF 方法训练的模型，发现两者对于不同标签的预测准确度有一定的倾向，总的来说 CRF 对于新文本的预测效果优于 LSTM-CRF。提取两者标记为基因和蛋白的词汇汇总，对基因进行富集，对蛋白质绘制互作网络，发现宫颈癌相关基因在分子功能方面主要负责 binding 和催化活性，在生物进程中主要负责细胞转化，生物调节以及代谢等，在细胞组成方面主要是在细胞解剖体中。TP53 蛋白是宫颈癌相关蛋白互作网络中的一个重要节点。

对 AGAC 数据库的序列标注任务与新文本挖掘

吴思琪 [1], 苏馨如 [2]

[1] 华中农业大学信息学院，430070，武汉，湖北，中国
[2] 华中农业大学信息学院，430070，武汉，湖北，中国

摘要

AGAC 做为主攻基因突变与疾病之间的关系的语料库，目前已按照不同级别的语言信息、语义信息完成对相关文本的标注，是良好的参考与实验材料。

CRF 做为神经网络出现前最好用的标注方法，拥有成熟的标注机制。在本实验中，我们将 CRF 分别与神经网络模型 LSTM 和 BERT 相结合，形成 LSTM-CRF 和 BERT-CRF 方法，将 AGAC 中的数据文件分为测试集和训练集，使用训练集分别对上述模型进行训练，并用测试集对训练得到的模型模型的标注效果分别进行评估。我们将 LSTM-CRF 和 BERT-CRF 方法训练模型的标注效果与 CRF 标注的效果进行对比，发现 LSTM-CRF 的标注结果与 CRF 差不多，BERT-CRF 方法训练模型的标注效果则要显著高于他们。

最后我们使用 1045 篇宫颈癌文献的标题与摘要做为预测文本，代入 LSTM-CRF 和 CRF 方法训练的模型，发现两者对于不同标签的预测准确度有一定的倾向，总的来说 CRF 对于新文本的预测效果优于 LSTM-CRF。提取两者标记为基因和蛋白的词汇汇总，对基因进行富集，对蛋白质绘制互作网络，发现宫颈癌相关基因在分子功能方面主要负责 binding 和催化活性，在生物进程中主要负责细胞转化，生物调节以及代谢等，在细胞组成方面主要是在细胞解剖体中。TP53 蛋白是宫颈癌相关蛋白互作网络中的一个重要节点。

关键词: CRF, LSTM, BERT, 宫颈癌

1 课题概况

随着互联网的发展，可以搜索到的文本呈几何倍数上涨，如何从海量的文本中筛出我们需要的信息，是当下生物信息乃至各行各业的热点问题。假期中我们参与了课外比赛，赛程中尝试绘制了词云和利用了神经网络，一是引起了我们对于自然语言处理的兴趣，二是发现自己学习语言处理仍是有一定难度，希望可以通过课程更系统地学习一下，本着挑战一下自己的原则，选择了课题四。

在本文中，因为 AGAC 语料库的注释已相对成熟，我们更希望通过 AGAC 训练出来模型对新文本进行挖掘，所以我们使用 1045 篇宫颈癌相关的文本作为待预测集，希望对预测出来的基因与蛋白做分析。

2 数据

本文中我们将对 AGAC 语料库中的文本进行序列标注，AGAC 语料库聚焦于人类突变基因所导致的功能变化。该语料库将数据划分为训练集和测试集 [1]，其中训练集包含 250 篇文献的标题与摘要，测试集包含 1000 篇文献的标题与摘要，数据文件的格式有.json、.tsv、.txt，分别记录了文献的相关信息。本文所有使用数据均来自于`http://pubannotation.org/collections/AGAC`与`https://github.com/bionlp-hzau`两个网站。

而关于后面将用于预测的有关宫颈癌文本的获取，我们是在 Pubmed 中检索“cervical cancer hpv mutation”并利用 Pubtator 下载了所有相关的 1045 篇文献的标题与摘要。

3 研究方法

3.1 研究方法的算法背景，与其他方法的联系与区别

序列标注是一个比较基础的 NLP 任务，它涵盖的范围非常广泛，可用于解决一系列对字符进行分类的问题，而本文中需要完成的是对 AGAC 语料库的 BIO 标注。

要完成序列标注任务则需要构建相应的模型，相关的模型一般可分为生成式模型和判别式模型，若我们用 $(x_1, x_2...x_m)$ 表示模型输入的观测序列 X，一般为词序列，而对应的标记序列 $(y_1, y_2...y_m)$ 则是标注模型输出的状态序列 Y。其中生成式模型是对联合概率 $p(X,Y)$ 进行建模，而判别式模型则是对条件概率 $p(Y|X)$ 进行建模, 比较适合做 Decoding 问题。较早提出的隐马尔可夫模型 HMM 就是一种生成式模型，后面提出的最大熵马尔可夫模型 MEMM 和条件随机场 CRF 都是判别式模型。本文所要完成 BIO 标注任务就是利用 CRF 模型进行标注，即 Decoding 问题的解决。

对于普通的齐次一阶 HMM 模型，由于状态转移概率与输出概率的设置导致观测序列 X 的元素之间存在独立性，而显然对于一个文本序列来说，上下文之间一定存在某种联系，所以这种假设在此处是不合理的。MEMM 模型的提出则解决这一问题，即 MEMM 并没有像 HMM 通过联合概率建模，而是直接学习条件概率 $p(Y|X)$，这就使得 y_t 同时受 y_{t-1} 与 x_t 的影响，从而使得 x_{t-1} 与 x_t 之间不再具有独立性。又因为 MEMM 是利用最大熵模型来学习条件概率的，所以可推导出其概率公式。

但 MEMM 模型由于沿用了马氏链，所以导致存在标注偏差问题，即当状态转移概率分布的熵越小时，观测序列 x_t 对 y_t 的影响权重就会越小，从而导致对 y_t 的预测存在偏差。而 CRF 模型的提出就解决这一标注偏差问题，其实质是将有向的马氏链转化为无向图模型，利用随机场的因子分解可推导出 CRF 模型的条件概率为：

$$p_{\vec{\lambda}}(\vec{y}|\vec{x}) = \frac{\exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i f_i(y_{i-1}, y_i, \vec{x}, j)\right)}{\sum_{\vec{y}\in Y}\exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m}\lambda_i f_i(y_{i-1}, y_i, j)\right)} \tag{1}$$

其中，λ 表示学习参数，而特征函数 f 是 y_{t-1}、y_t 和 X 的函数，根据其无向图模型可对其进行分解为：

$$f(y_{t-1}, y_t, \vec{x}) = \Delta(y_t, \vec{x}) + \Delta(y_{t-1}, y_t, \vec{x}) \tag{2}$$

前一项可认为是状态函数，后一项可认为是状态转移函数。说这明了 CRF 不仅可以学习观测序列与状态序列之间的联系，也可以学习到状态序列各元素之间的联系。对应到文本标注中，即利用 CRF 模型既可以通过学习单词之间的特征来预测标签，也可以通过标签之间的转移变化情况来进行标签的预测。

3.2 研究方法中的核心思路

3.2.1 LSTM+CRF 模型

CRF 模型之后，神经网络开始兴起，凭借其优良的非线性拟合能力，开始被广泛被应用于文本挖掘领域。其中，长短期记忆 (LSTM) 是一种特殊的 RNN，主要是为了解决长序列训练过程中的梯度消失和梯度爆炸问题，相比普通的 RNN，LSTM 能够在更长的序列中有更好的表现。

LSTM 与传统 RNN 的不同在于节点结构的复杂化，传统 RNN 对节点只有 1 个输入函数，且为 x_t 与 h_{t-1} 的函数；而 LSTM 对节点有 4 个输入函数，均为 x_t、c_t 与 h_{t-1} 的函数。LSTM 节点的具体结构如图1A 所示，c_t 与 h_t 共同作为节点的转移状态 $(h_t = \sigma_h(c_t))$，整个节点通过四个激活函数控制状态的转移与输出，输入门控制 z^i 是否需要被输入，遗忘门控制 c_t 是否需要被更新，输出门控制 h_t 是否需要输出。

为了更好的预测结果，LSTM 模型往往需要构建多层即 Multiple-layer LSTM，若需要通过上下文同时训练模型，可以构建双向模型即 Bi-LSTM。

加 CRF 层的原因：由于本文实现的任务为序列标注，所以在 LSTM 后需要在添加 CRF 层，这是因为 LSTM 只能获得输入序列的特征，而无法获得预测标签之间的特征，而在上文我们已指出 CRF 模型可以获得标签转移的特征，所以更有利于结果的预测。

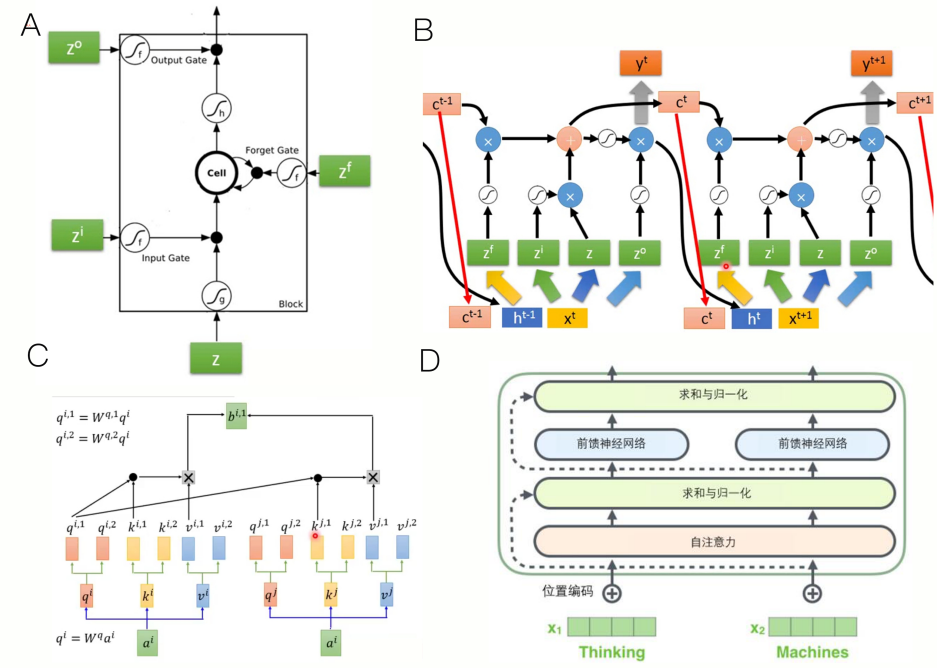


图 1: 模型示意图。图片来源：`https://b23.tv/rLaUrT`

3.2.2 BERT+CRF 模型

关于 BERT 的模型架构，则是直接采用了 2017 年论文中的 multi-layer bidirectional Transformer 编码器部分 [2]。每个解码器都可以分解成两个子层，由图1D 所示，第一层由 Multi-Head Attention 层与一个残差模块构成，第二层由普通的前馈神经网络与残差模块构成。

Self-Attention 与 Multi-Head Attention：由于 RNN 或 LSTM 的训练需要根据文本顺序依次读入进行参数训练（若为双向模型则需要从两个方向各训练一次），所以很难设置并行来提高训练效率。所以，自注意力机制（Self-Attention）的提出则解决了这一问题。

自注意力机制基本结构如下：输入序列单词 x_i 首先通过词嵌入获得 $a_i(a_i = Wx_i)$，然后通过与三个权重矩阵相乘获得生成三个向量，即一个查询向量 q_i、一个键向量 k_i 和一个值向量 v_i 。接着，对向量 q_i 与 k_i 进行 Scaled Dot-Product Attention 的计算获得向量 $\alpha_i(\alpha_{i,j} = q_i \cdot k_j / \sqrt{dim_{q,k}})$，向量 α_i 经过计算 softmax 得到 $\hat{\alpha}_{i,j}(\hat{\alpha}_{i,j} = \exp(\alpha_{i,j}) / \sum_j \exp(\alpha_{i,j}))$，最后与向量 v_i 相乘并求和得到输出向量 $b_i(b_i = \sum_i \hat{\alpha}_{i,j} v_i)$。

而 Multi-Head Attention 则是在 Self-Attention 的基础上，对每个查询向量、键向量、值向量进行再次通过与权重矩阵相乘形成多个查询向量 q_i、键向量 k_i 与值向量 v_i，意在学习更多的序列特征。最后的 b_i 可由一个权重矩阵 W_0 乘以 $(b_{i,1}, b_{i,2} \cdots)^T$ 获得, 如图1C 所示。

由上述过程可知，无论是 Self-Attention 还是 Multi-Head Attention，每个单词的输入都会学习整个文本序列的信息，所以计算上不再受上下文的束缚可以方便地进行并行化。

位置编码 Position Encoding：由上述内容可知 Multi-Head Attention 是不具有文本中单词的位置信息的，所以要通过位置编码 e_i 与词嵌入后的 a_i 相加作为后面的输入（$a_i = a_i + e_i$）。

残差模块 Add+Norm：这部分的操作是将带有位置信息的输入 a，输出 b 相加（防止梯度弥散），再进行 Layer Norm，即将每个 Batch data 进行标准化，使得 Batch data 的均值 μ=0，标准差 σ=1。

BERT 模型的创新主要在其预训练过程，它包括两个任务 [3]：

Masked LM(MLM)：为了训练一个可以双向学习文本信息的模型，BERT 采用了一种方法是随机屏蔽（masking）每个序列中 15% 的单词，然后只预测那些被屏蔽的单词。在大量文本的训练之后，BERT 模型使得具有相似语义的单词获得相近的词嵌入，从而服务于接下来具体的训练任务。

下一句预测：此任务是预测两个句子是否是连续的，通过引入两个特别的 token，即 [SEP] 与 [CLS], [SEP] 代表两个句子连接的位置，[CLS] 代表预测的结果，一般放在句子开头。

综上，BERT 模型是一种预训练模型，它可以将我们输入的文本进行更好更符合语义地编码，然后根据不同的文本分析任务连接不同的神经网络层，而在本文中我们要完成序列标注任务，所以需要在 BERT 模型后接上 CRF 层。

3.3 本文的方法部分与课堂讲授内容的联系和与补充

Softmax 函数: 对于多分类问题，神经网络的输出层通常为 Softmax 函数，其定义为：

$$\sigma_i = Soft\max\left(z_k\right) = \frac{\exp\left(z_k\right)}{\sum_{c=1}^{C}\exp\left(z_c\right)} \tag{3}$$

其中，z_c 表示输出层第 c 个节点的输出值，C 表示输出节点的个数，即为分类标签 k 的个数。通过 Softmax 函数就可以将多分类的节点输出值转换为范围在 [0, 1] 且 sum 为 1 的概率分布。

而根据之前 MEMM 与 CRF 模型的推导结果可知，Softmax 函数也可计算条件概率：

$$p\left(y|x\right) = \frac{\exp\left(f_y\right)}{\sum_{c=1}^{C}\exp\left(f_c\right)} = Soft\max\left(f_y\right) \tag{4}$$

其中，f 即为特征函数。由于序列标注本质也是一种多分类任务，所以这也解释了 CRF 模型可以直接作为神经网络的输出层的原因。

Loss 函数: 在模型学习的过程中，需要根据 Loss 函数不断优化特征函数的各项参数与神经网络节点的各项权重，在多分类任务中则是计算交叉熵，其表达式为：

$$L\left(w\right) = -\sum_{i=1}^{m} y_{i(k=j)}\ln\left(\sigma_i\right) \tag{5}$$

其中，j 为样本 i 所对应的真实标签的编号，k 为预测标签，σ 为 Softmax 函数值。

4 算法实践和代码编写要求

4.1 任务描述

BIO 标注是指将每个元素标注为“B-X”、“I-X”或者“O”。其中，“B-X”表示此元素所在的实体属于 X 类型并且此元素在此实体的开头，“I-X”表示此元素所在的实体属于 X 类型并且此元素在此实体的中间位置，“O”表示不属于任何类型。

针对 AGAC 语料库，我们需要 CPA，Disease，Enzyme，Gene，Interaction，MPA，NegReg，Pathway，PosReg，Protein，Reg，Var 这 12 个实体的信息，根据 BIO 标注的特点则一共是 25 个标签类别。

根据上文对各类模型的解读与分析，我们将利用 AGAC 语料库中已经标注好的训练集和测试集文本，构建 CRF、LSTM-CRF、BERT-CRF 模型并进行模型训练与模型的测试评估。然后对结果进行比较分析。最后，我们根据 AGAC 语料库的特点，准备利用上述的三个已被训练好的模型对宫颈癌相关文献的文本信息进行挖掘。

4.2 实验设计

4.2.1 序列标注

CRF 模型的构建我们利用的是 Wapiti。具体流程为：下载 Wapiti 的安装包，配置工作环境。利用特征函数文件对 AGAC 语料库中的训练集文件进行训练，得到模型，将模型存放入 AGAC_train.mod 中。运用训练出来的模型对测试集中的文件进行预测，将所得结果存放入 train_out.tab 中。对于预测出来的结果，针对 12 个实体，从准确率（P 值），召回率（R 值），FB1（F 值）三个方面进行打分，将结果输入到 train_out.eval 中。

LSTM-CRF 模型与 BERT-CRF 模型的建立与评估大致和上述过程相同，但这两个模型的代码实现均是基于 python3.7 与 torch1.7.1 完成的。

4.2.2 新文本挖掘

利用 EDirect 寻找宫颈癌相关的文章的 PMID，使用 PubTator 获取相应文本。本文摘取相关的 1045 篇文献的标题与摘要进行后续分析。

首先我们利用 Wapiti 进行新文本预测。我们对这 1045 个文本进行格式转换，使其成为 Wapiti 模型接受的 BIO 格式。用建立好的 AGAC_train.mod 中的模型对测试集中的文献进行预测，并对预测结果进行生物学分析。

然后，利用同样的 1045 个宫颈癌相关的文本建立预测数据集（需要格式转换），再用 LSTM-CRF 建立出来的模型对其进行预测，并对两者的预测结果进行比较，最后对预测结果进行生物学分析。

4.3 关键代码

详见`https://github.com/bionlp-hzau`中的 LSTM_CRF_useAGAC 项目与 BERT-CRF-for-BioNLP-OST2019-AGAC-Task1 项目。

5 主要的生物信息学实验和实验结论

5.1 针对 AGAC 数据库的序列标注

5.1.1 模型训练结果对比

以 10 个 epoch 为步长，记录 LSTM-CRF 训练的模型的损失值（Loss 值）以及 BERT-CRF 训练的模型的损失值，如图2所示，LSTM-CRF 训练的模型的损失值以及 BERT-CRF 训练的模型的损失值都呈现先急速下降，后缓慢下降，最终稳定在无限趋近于 0 处。表明 LSTM-CRF 与 BERT-CRF 的训练方向都是正确的。BERT-CRF 训练的模型的损失值收敛速度略快于 LSTM-CRF 的模型，说明 BERT-CRF 训练的模型，尤其是前 20 轮的模型，效果要优于 LSTM-CRF 的。

5.2 模型评估结果对比

以 10 个 epoch 为步长，记录 Wapiti 训练的模型的 FB1 值为 20.01，LSTM-CRF 训练的模型的 FB1 值以及 BERT-CRF 训练的模型的 FB1 值（表1），发现 LSTM-CRF 训练的模型的标记效果相对于 Wapiti 没有显著的提升，两种方式训练出来的模型都处于效果一般的状态。而 BERT-CRF 训练的模型标记效果相对于 Wapiti 有显著提升，FB1 值基本可以稳定在 50 左右。认为 BERT-CRF 训练的模型是最可信的模型。

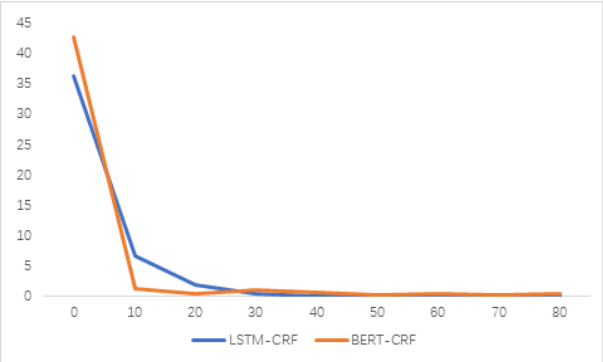


图 2: 训练过程 Loss 图。图片来源：Excel 绘制

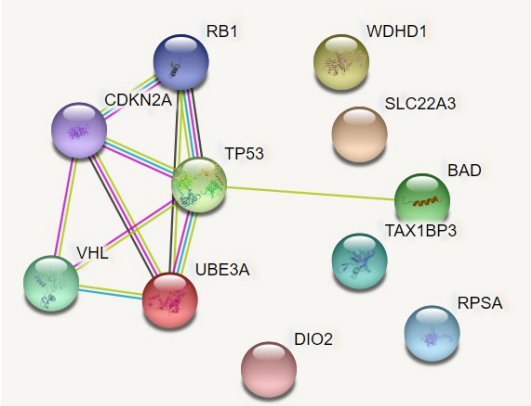


图 3: 蛋白互作图。图片来源：STRING 网站绘制

表 1: LSTM-CRF 与 BERT-CRF 模型的 FB1 值对比

epoch	LSTM-CRF	BERT-CRF
10	14.05	49.89
20	19.54	51.7
30	16.51	51.72
40	18.71	50.73
50	20.02	52.33
60	20.97	51.43
70	21.19	49.92
80	20.12	52.57

5.3 针对宫颈癌的文本挖掘

对于 Wapiti 与 LSTM-CRF 模型预测出来的文本我们主要进行了对比分析，并进行对基因的文本进行富集分析，然后筛选出被预测为蛋白的文本进行互作网络的绘制，得到其生物学意义。

在预测新文本的过程中，Wapiti 建立的模型共标记处是基因的词组 132 个，实际是基因的仅有 28 个，准确率为 21.2%。而 LSTM-CRF 建立的模型共标记出了 343 个基因，而真正是基因的仅有 22 个，准确率为 6.4%。在标记蛋白的过程中，LSTM-CRF 共标记了 121 个蛋白，其中是真正的蛋白的为 2 个。而 Wapiti 建立的模型一个蛋白也没识别出来。可见 LSTM-CRF 建立的模型标注效果差，不适用于此类文本分析。它们对于蛋白的灵敏度都不高，也符合前文中在测试集上评估的结果。

综合两种模型标记出来的基因，我们对于该基因集进行富集分析。结果表明，宫颈癌相关基因在分子功能方面主要负责 binding 和催化活性，在生物进程中主要负责细胞转化，生物调节以及代谢等，在细胞组成方面主要是在细胞解剖体中。

综合两种模型标记出来的蛋白，我们利用网页版对其进行蛋白互作分析 (`https://string-db.org/`)，得到蛋白质互作网络（图3）。

6 后记

6.1 课程论文构思和撰写过程

关于课程论文，我们最初的构想是宏大的，我们要完整地学习一遍神经网络，自己写代码训练模型。后来确实发现内容很多，时间不够。所以就想着利用课上给定的模型完成后面的训练测试，然后找新文本，对新文本进行预测，对比三种模型的预测结果，寻找生物学意义。当时确立方案的时候，我们还想这么做会不

会太简单了，因为感觉也不用自己写很多代码。

但其实还是碰到了不少困难，首先是电脑的环境问题，之前使用服务器的机会不是很多，这次课也是我们第一次在自己电脑上安装和使用 Ubuntu 系统，用过之后发现还挺方便的。解决完电脑环境问题，就是训练模型和预测新文本了，训练模型就是按照师兄师姐给的代码跑一遍，就是训练时间长一点，特别是 BERT-CRF 模型，我们的电脑配置需要跑一天多，而且我感觉数据量也不是那么大，此刻才认识到深度学习为啥需要 GPU 了，pytorch 框架也是为了构建张量方便在 GPU 上跑。

接下来是预测新文本，我们研究了一下代码，觉得直接改新文本的格式比较方便，这样就不用修改老师给的代码，因为是预测所以自己设的标签并不会被程序使用。然后我们就这么进行了，碰到的第一个困难就是文本的处理，正则表达式真的有点累人，而且文本中本来也存在着一点错误，比较印象深刻的问题是句点的处理，它可能是小数点，也可能是基因或蛋白中的句点，还可能是网站中的，总之想了很久，最终是查网上的代码加自己手动改正一些解决的。

得到每个模型对应的新文本格式之后，接下来就是预测啦，Wapiti 和 LSTM-CRF 虽然不是一次就成的，但都问题不大，都顺利预测成功，但在跑 BERT-CRF 的预测上，却把我们难倒了，首先在老师提供的代码中好像没有找到预测的函数文件，所以我们第一反应就是自己写，首先我们将训练出来的最佳模型参数文件 pkl 文件通过 torch 导入，先用 test 数据集验证了一遍模型效果，但发现预测结果极其地差，后来在网上查了很多，我们觉得自己的思路没有错，但显然不能用这个模型去预测新文本。然后修改了一下代码在 jupyter 中将模型又跑了一遍，从而保存那个最佳的模型参数，结果确实是这样，然后我们先是将原有的代码做了修改，来预测新文本，但是却是一直报错。无奈之下，我们去仔细研究了一下模型原理，发现自己的思路有问题，所以根据新的理解我们调整了预测的代码，本来觉得信心满满，但是却是提示内存不足 buy a new RAM，当时就有点丧气，因为快要到截止日期了，所以我们开启了 Plan B，我们决定一个人继续用 BERT-CRF 做新文本的预测，一个人对新文本预测的结果进行分析，寻找一些对模型的评价乃至生物学意义。本以为是一个纯神经网络的课题，没想到最后还是扣回了生信的老本行。需要预测的实体有 12 个，有一些因为与一些大众的简写重复了，很难找到确切的定义 (像 CPA)，从而最后决定只分析被标为基因和被标为蛋白的。基因这学期在自然语言处理中我们学习了用 R 包做基因富集，刚好在这可以有用武之地，温故而知新，并借着这个机会仔细看一看每个结果的意义。蛋白我们这学期学习了系统与合成生物学，之前一直在说蛋白互作网络，所以也选择分析它，温习了一下当时听的头晕的连接度什么的。处理文本用了之前 perl 的知识，大三基本都在调包，太久不写基础代码了，简单地分析个文本都有点吃力了。感谢 perl 老师保留课程群及课件永久可下载，助力了本次的数据处理。

很遗憾这次因为时间限制没能把课程论文做到我们想象中那么好，希望通过未来长期的学习可以让我们曾经看似飘渺的凌云壮志都得以实现。这次实验，现学的东西没用上，之前的知识反而起了作用。一切在冥冥之中自有回响，感受到了不同的学科融合在一起解决问题的奇妙，是我们本次课程论文最大的收获。

6.2 所参考主要资源

CRF 模型：`https://b23.tv/gMjgfS`

Self-Attention：Attention is all you need.[2]

RNN、LSTM、BERT 模型 [3]：`https://b23.tv/rLaUrT`

代码参考：`https://github.com/bionlp-hzau/`

6.3 人员分工

吴思琪：模型原理部分撰写，代码实现

苏馨如：模型结果与预测部分撰写，代码实现

参考文献

[1] Yuxing Wang, Kaiyin Zhou, Mina Gachloo, and Jingbo Xia. An overview of the active gene annotation corpus and the bionlp ost 2019 agac track tasks. In *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks*, pages 62–71, 2019.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *arXiv*, 2017.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

8.2 江毅炜《针对 AGAC 数据库的序列标注》

命名实体识别 (NER) 是自然语言处理 (NLP) 的基本任务之一，是一个分类任务，被称为序列标注任务，即文本中不同的实体对应不同的标签。解决此类问题的方法经历了早期的基于规则或基于字典方法; 传统的机器学习方法，例如 HMM，MEMM，CRF; 深度学习的方法，例如 RNN-CRF、CNN-CRF; 以及近期的基于注意力模型的 BERT 模型。为了解决 AGAC 数据库序列标注的任务，将从 CRF，LSTM，BERT 算法入手，并用 pytorch 将算法实现，搭建 CRF、LSTM-CRF、BERT-CRF 三种网络，并用这三种网络分别对 AGAC 数据库进行训练，比较 BERT-CRF、LSTM-CRF 训练的时间，用损失函数量化每次训练的效果，发现 BERT-CRF 训练的时间远高于 LSTM-CRF 训练的时间，但是 BERT-CRF 每轮训练的效果要更好。最后用 LSTM-CRF 训练 AGAC 数据库后的模型对 1985 篇有关乳腺癌摘要文本进行有关基因和突变的数据挖掘，经过筛选，发现了 10 个与乳腺癌相关的基因和 5 个与乳腺癌相关的突变。

针对 AGAC 数据库的序列标注

江毅炜 [1]

[1] 华中农业大学信息学院，430070，武汉，湖北，中国

摘要

命名实体识别（NER）是自然语言处理（NLP）的基本任务之一，是一个分类任务，被称为序列标注任务，即文本中不同的实体对应不同的标签。解决此类问题的方法经历了早期的基于规则或基于字典方法；传统的机器学习方法，例如 HMM，MEMM，CRF；深度学习的方法，例如 RNN-CRF、CNN-CRF；以及近期的基于注意力模型的 BERT 模型。为了解决 AGAC 数据库序列标注的任务，将从 CRF，LSTM，BERT 算法入手，并用 pytorch 将算法实现，搭建 CRF、LSTM-CRF、BERT-CRF 三种网络，并用这三种网络分别对 AGAC 数据库进行训练，比较 BERT-CRF、LSTM-CRF 训练的时间，用损失函数量化每次训练的效果，发现 BERT-CRF 训练的时间远高于 LSTM-CRF 训练的时间，但是 BERT-CRF 每轮训练的效果要更好。最后用 LSTM-CRF 训练 AGAC 数据库后的模型对 1985 篇有关乳腺癌摘要文本进行有关基因和突变的数据挖掘，经过筛选，发现了 10 个与乳腺癌相关的基因和 5 个与乳腺癌相关的突变。

关键词：CRF，BERT，LSTM，乳腺癌，AGAC 数据库

1 课题概况

生物信息处理的数据多数为文本文件，其中包括以 A、T、G、C 四种碱基组成的测序文件，各种生物数据库中的文本文件，基因组、转录组等数据，如何从中获取有用的信息是做进一步研究的关键，本课程《生物文本挖掘与知识发现概论》是针对生物文本进行挖掘从而获得知识的发现，必然有可以学习的数据处理，信息挖掘的方法，可以拓展我们对测序基因数据处理的思路，并且可以对基因组、转录组等数据进行检验或补充，例如将生物文本挖掘的基因与 GWAS 中分析出基因进行比较。

在文本挖掘的技术中，序列标注是获取有效信息的方法，即设输入的序列为 $X = (x_1, x_2, ..., x_n)$，输出的序列为 $Y = (y_1, y_2, ..., y_n)$ 将文本与标签一一对应。CRF、LSTM、BERT 是处理序列标注任务的不同模型，将从算法出发理解三种模型，并用 pytorch 工具完成对 CRF、LSTM-CRF、BERT-CRF 三种网络的搭建，分别对 AGAC 数据库进行序列标注，通过训练的时间和训练的效果比较三种模型的差异，最后从 pubtater 获得 1985 篇与乳腺癌有关的摘要，用 LSTM-CRF 训练得到的模型进行序列标注，挖掘与乳腺癌有关的基因和突变。

2 数据

AGAC 数据库是一个由人类专家注释的语料库，目的是捕捉致病背景下突变基因的功能变化 *[1]*，数据来源于 pubannotation 这个网站，包括了 AGAC-sample，AGAC-test，AGAC-raining，三部分一共 1300 篇摘要[1]。由于 AGAC 数据库的数据主要与基因突变，致病有关，所以对于新文本的预测选择与这两方面有关的数据即乳腺癌，用 edirect 工具从 pubtater 获取 1985 篇与乳腺癌摘要。三种网络对数据库的训练数据都是以“单词　标签”模型的文本数据，即实体与标签一一对应。

[1] http://pubannotation.org/collections/AGAC

3 研究方法

3.1 研究方法的算法背景，与其他方法的联系与区别

HMM（隐马尔科夫模型）是一个生成模型 ($\gamma(\pi, A, B)$, 其中 π：初始状态概率分布；A：状态转移矩阵，B：发散矩阵)，HMM 有有两个假设，即齐次一阶马尔可夫假设和观测独立假设，其中齐次是具有相同的状态转移概率矩阵，与时间无关，一阶为当前状态与前一个状态有关。但是这两种假设都不是很合理，而 MEMM 模型打破了观测独立假设，是一个判别模型，对于链式结构例如标注问题，判别模型即比 HMM 要更好，但是在局部状态转移的时候，需要归一化，引起了 Label Bias Problem（标注偏置问题），转移概率分布熵越小，越不会关注输入 X_i。CRF（链式随机场，下同）是由 MEMM 由有向图转变为无向图的改进模型“如图 *1*所示”，是一个判别式模型，克服了 Label Bias Problem，由 MEMM 的局部归一化变为了全局归一化，并且打破了齐次一阶马尔可夫假设。

传统的神经网络只能一个个单独的处理输入，而前后输入是完全没有关系的，在 NLP 领域中，输入单词时常用词嵌入，将词汇转换为词向量，输入到神经网络，但是在语序上很难处理，RNN（循环神经网络）就是为了解决语序问题而出现的模型，RNN 的模型为 $h_{i+1} = sigmoid(W * h_i + U * x_i + b)$，其中 x_i 表示在第 i 步输入的单词，h_i 表示当前的隐藏层权值向量，h_{i+1} 表示输出，W 表示权重，因为 RNN 有“梯度消失”和“梯度爆炸”的现象，其中梯度消失意味着 RNN 对长距离语义的捕捉能力失效了。LSTM（长短期记忆人工神经网络）解决了上述两个问题，与 RNN 相比多了三个控制结构“如图 *1*所示”，分别为：遗忘门（forget gate）控制上一个时刻状态的多少保留到当前时刻；输入门（input gate）控制当前时刻的输入有多少保存到当前状态；输出门（output gate）控制当前状态有多少进行输出。LSTM 在 RNN 基础上增加了对过去状态的过滤，选择出一些状态对当前更有影响而不是选择最近的状态。

BERT(Pre-training of Deep Bidirectional Transformers for Language Understanding) 是一个用 Transformers 作为特征抽取器的深度双向预训练语言理解模型。其中核心是基于多个 Transformers 的 encoder 和 Attention 机制 *[2]*。从架构图型“如图 *2A* 所示”中可以看到, BERT 分三个主要模块.，分别为 Embedding 模块（黄色），Transformer 模块（蓝色），预微调模块（绿色）。Embedding 模块由三种 Embedding“如图 *2C* 所示”求和而成，包括 Token Embeddings（词向量）、Segment Embeddings（单词属于哪个句子）、Position Embeddings（学习出来的向量）。Transformer 是一个 encoder-decoder 的结构“如图 *2B* 所示”，由若干个编码器和解码器堆叠形成，在 decoder 中，在训练的过程中并不输入所有的单词，而是遮住一部分单词并用已经训练的模型对遮住的单词进行预测并且进行模型的修正。

3.2 研究方法中的核心思路

深入了解 CRF、LSTM、BERT 算法，用 wapiti、pytorch 工具对上述算法进行模型的构建，用构建的 CRF、LSTM、BERT 网络对 AGAC 数据库进行训练，比较在训练相同次数时的损失函数检验每次预测的效果和运行的时间，最后用 LSTM-CRF 对 AGAC 数据库训练的模型对 1985 篇乳腺癌文本进行预测，进行基因和变异相关数据的挖掘，用挖掘的出来的数据进行 GO 富集分析，将挖掘出来的数据做进一步的功能分析。

3.3 本文的方法部分与课堂讲授内容的联系和区别/或补充

利用课堂讲授代码做不同模型之间的比较，并用已经训练出来的模型做进一步对其他文本的预测，挖掘新的数据，做进一步分析。

[2]CRF 和 MEMM 图片来源：https://www.alibabacloud.com/blog/hmm%2C-memm%2C-and-crf%3A-a-comparative-analysis-of-statistical-modeling-methods_592049

LSTM 图片来源：https://en.m.wikipedia.org/wiki/Long_short-term_memory

[3]图片来源：https://www.cnblogs.com/rucwxb/p/10277217.html

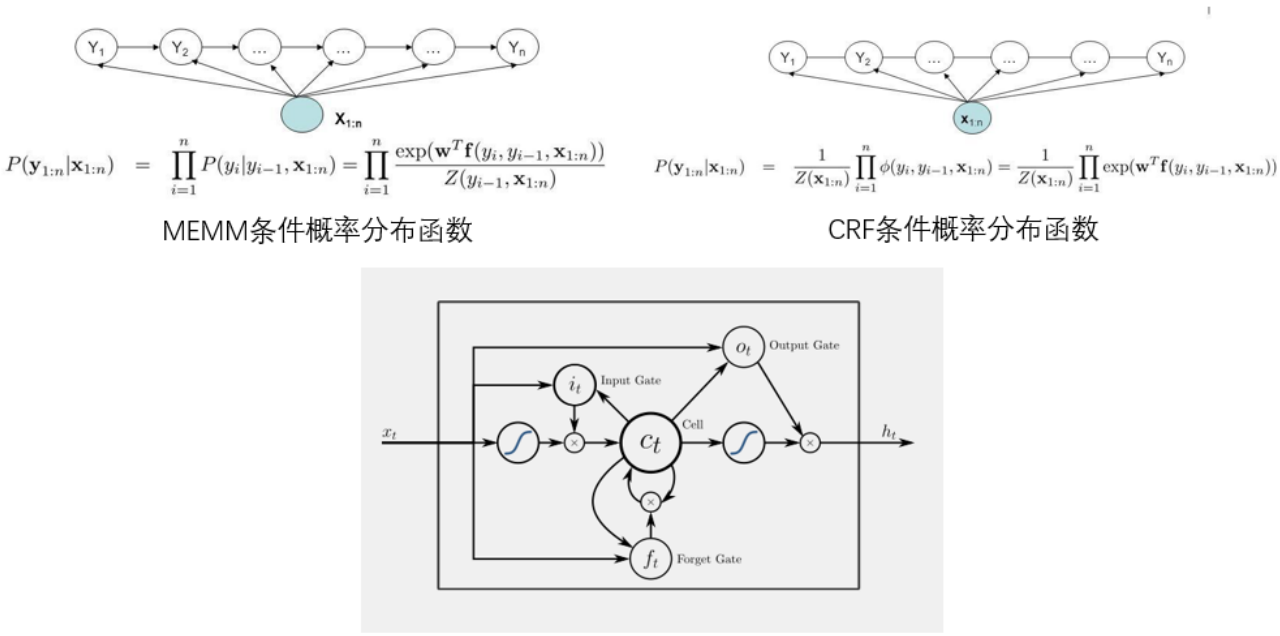


图 1: CRF、MRMM 图模型和条件概率分布函数、LSTM 神经网络[2]

4 算法实践和代码编写要求

4.1 任务描述

在用 wapiti 工具[4]构建 CRF 训练和预测 AGAC 数据库时，只需要按照手册修改相应的参数，在用 pytorch 构建 BERT-CRF 网络时，BERT 由 12 层 Encoding 层和一层 mash 层构成，其中 12 层 Transformer Encoder 都是为了将输入的序列找到 BERT 的特征并且作为该任务的输入，attention_mask 层是在训练的过程中并不输入所有的单词，而是遮住一部分单词，用已经训练的模型对遮住的单词进行预测并且进行模型的修正，找到单词之间的联系, 以训练出一个更强大的语言模型。

LSTM 是两层网络一个前馈神经网络和一个 LSTM 类型的 RNN 网络，BERT 和 LSTM 都会输出一个概率矩阵，并且对于具体的单词输出具体的标签，最后在 BERT 和 LSTM 加上一个 CRF 层是学习标签之间的关系，即”B-Disease”(疾病开始单词的标记) 后面只会出现“O”（不感兴趣的单词）或者“I-Disease”（(疾病中间单词的标记), 而不会出现“I-Pathway”（通路中间的单词）。在训练的时候返回损失函数，在不断迭代中不断修正模型中的参数，以降低损失函数提高模型的准确性。

4.2 实验设计

从 AGAC 数据库中获得相关摘要，将 AGAC 获得的摘要进行分词处理并且附上标签，数据集中分别由 250 篇训练集和 250 篇测试集组成，将处理好的数据放入到写好的模型中进行训练，并且评价模型质量。对于 1985 篇乳腺癌的摘要，也进行分词的处理并且都附上初始值 O，放入训练好的模型中，进行修正，得到预测的标签。通过预测的标签找到对应的实体，将实体中的基因进行 GO 富集分析，找到这些基因的一些功能。

[4]https://wapiti.limsi.fr/manual.html#patterns

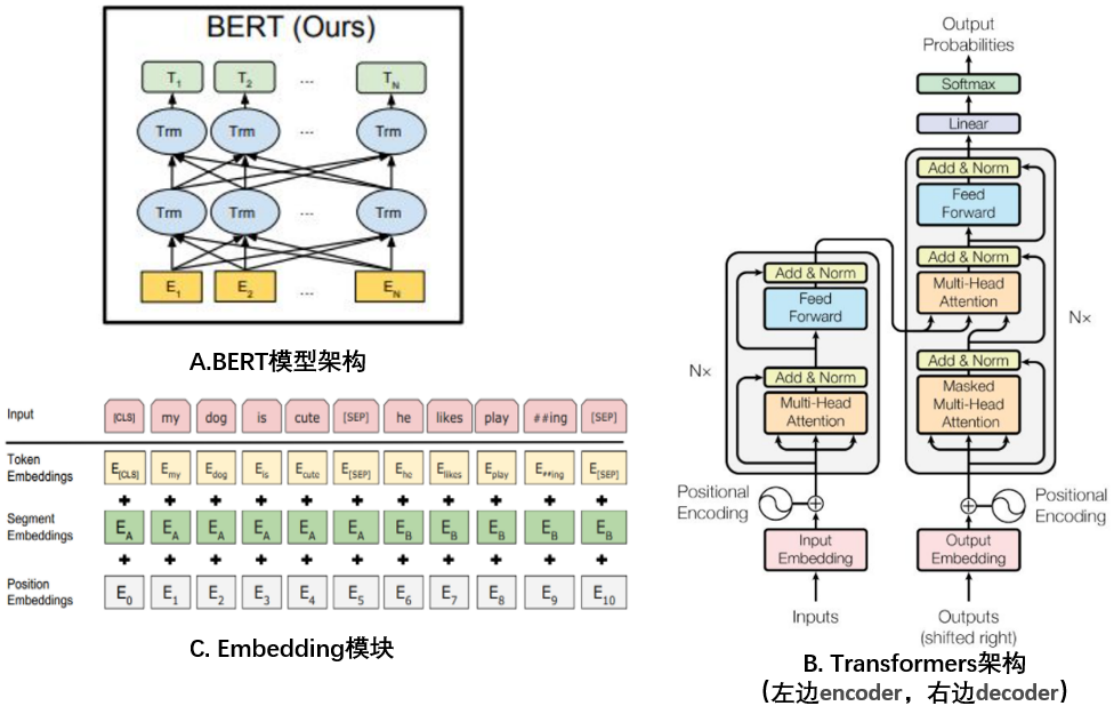


图 2: BERT 模型架构、Transformers 架构、Embedding 模块[3]

4.3 一些关键代码

```
代码来源：https://github.com/bionlp-hzau/BERT-CRF-for-BioNLP-OST2019-AGAC-Task1
https://github.com/bionlp-hzau/LSTM_CRF_useAGAC
# BertCRFTagger
class BertCRFTagger(nn.Module):

    def __init__(self, bert, hidden_size, num_tags, dropout):
        super().__init__()
        self.bert = bert
        try:
            self.crf = CRF(num_tags)
        except:
            self.crf = CRF(num_tags, batch_first=True)
        self.fc = nn.Linear(hidden_size, num_tags)
        self.dropout = nn.Dropout(dropout)

    def forward(self, input_ids, mask, tags=None):
        ......
#LSTM
class rnn(nn.Module):
    def __init__(self, cti_size, wti_size, num_tags):
    ......
        self.rnn = getattr(nn, RNN_TYPE)(       #RNN_TYPE = LSTM
            input_size = EMBED_SIZE,    #300
            hidden_size = HIDDEN_SIZE // NUM_DIRS,    # 1000/2
            num_layers = NUM_LAYERS,   #2
            bias = True,
            batch_first = True,
            dropout = DROPOUT,
            bidirectional = (NUM_DIRS == 2)
        )
```

5 主要的生物信息学实验和实验结论

用 wapiti 工具构建 CRF 训练和预测 AGAC 数据库后，准确率为 90.15%，但是结果具有一定的偏好性比如 NegReg 发现的很多，而 Enzyme 和 Protein 就很少甚至没有。在对 BERT-CRF 和 LSTM-CRF 训练的过程中，在每一轮 epoch 中 BERT-CRF 花费的时间远高于 LSTM-CRF，但是在单次训练的过程中训练效果要优于 LSTM，耗时更久是因为 BERT 有 12 层 Transformer，且最大长度为 128，在训练十次左右基本有了最优的模型，而 LSTM 虽然耗时较短但是需要训练二十次左右才基本有最优模型“如图 *3A* 和 *B* 所示”。

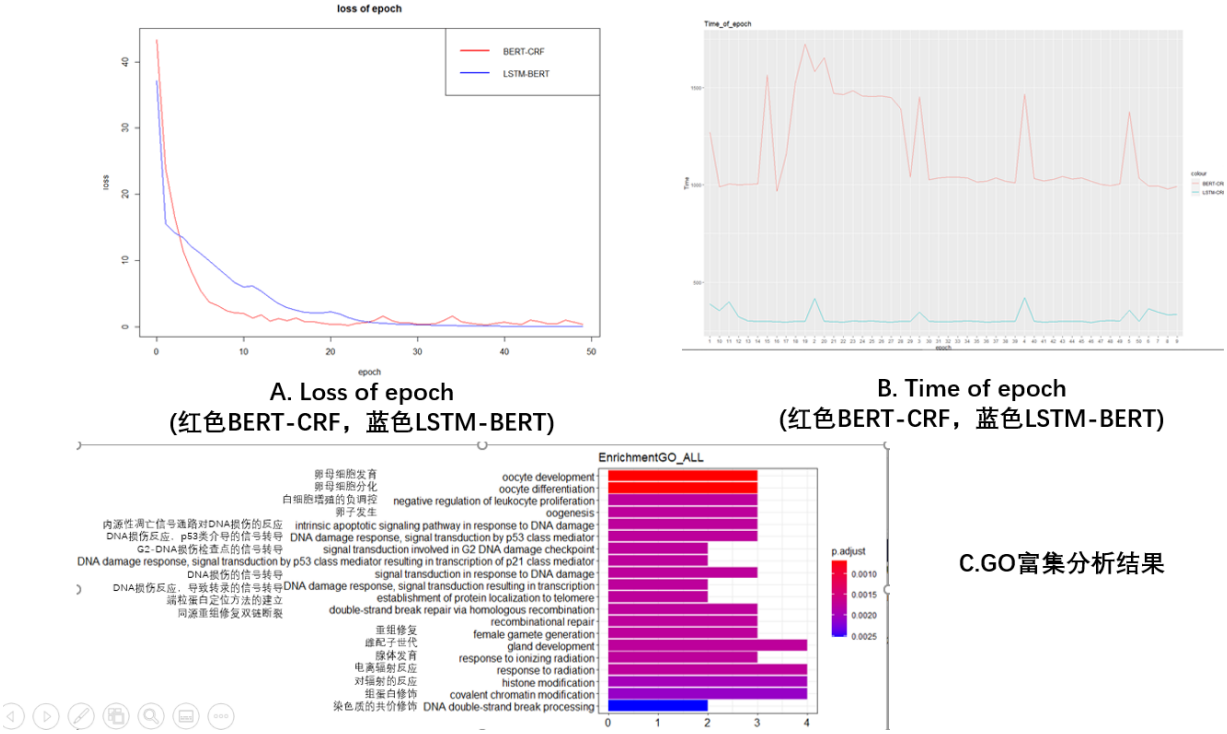


图 3: 训练时间和损失函数随训练次数的变化[5]

最后用训练二十次的 LSTM-CRF 模型对 1985 篇有关乳腺癌的摘要进行文本挖掘，最终标记了 290 个单词为 Gene 标签，120 个单词为 Var 标签，对这两类标签的单词进行提取，用 R 对其进行文本处理，即筛掉标点符号和数字等，最后进行人工筛选，选出来 10 个有意义的基因，“*表 1*中包含了与癌症和乳腺癌有关的基因（*BRCA1*、*BRCA2*、*PEG3*、*CEA*、*KMT2D*、*Cx43*、*AKT*）；与免疫调节有关的蛋白质和基因（*TNFAIP3*、*Th1*、*Th2*）；与糖类调节有关的蛋白（*UDP*、*Hex*）；检验 *DNA* 损伤的基因（*ATM*）；其他蛋白和基因（*TRPV1*、*HMGCS1*、*CRP*）”虽然在预测的结果中有一些没有用的信息，但是大大减少了人工筛选的文本量即（470000 左右单词到 400 左右单词），缩小了接近 1000 倍。

对于已经筛选出来的基因可以进一步假设，假设如下，上述基因中的抑癌基因突变并且检测 DNA 损伤的基因未能检测出抑癌基因的突变，从而导致细胞增殖失控称为癌细胞，从而影响了人体的免疫调节，由于细胞增殖需要消耗大量能量从而影响了糖类调节的基因，加快糖代谢提供能量，这些仅仅是基于这些基因和基因的功能做的生物学背景的假设，真正验证还需要生物学实验提供支撑，但是不可否认的是为生物学研究方向提供了一种可行性的思路。

将得到基因进行进一步的 GO 富集分析“如图 *3C* 所示，反复出现了 DNA 损伤、DNA 损伤修复、单例蛋白、组蛋白修饰、染色体共价修饰、p53 等词条，说明癌变和 DNA 的损伤有密不可分的关系，这点已

[5]图片来源：实验得到的数据用 R 作图

经在生物学上验证。同时出现的其他词条，例如卵母细胞，白细胞增殖，腺体发育等功能中可能存在与癌细胞相关的潜在关系。

表 1: 通过模型预测乳腺癌摘要，挖掘的与乳腺癌有关的基因和突变

Gene	Function	Mutation	Function
TRPV1	辣椒素受体	UDP	葡萄糖醛酸转移酶
BRCA1	遗传性乳腺癌有关的基因	Hex	己糖激酶
BRCA2	与乳腺癌有关的基因	CRP	C-反应蛋白
PEG3	一种癌基因	Th1	TH1 辅助细胞
HMGCS1	该基因的表达与肝脏生成胆固醇的相关	Th2	TH2 辅助细胞
ATM	与 DNA 损伤检验有关的一个重要基因		
TNFAIP3	TNFAIP3 基因突变会导致免疫系统持续运转		
CEA	癌胚抗原		
KMT2D	甲基转移酶 KMT2D 在肿瘤中作为异常的修饰因子		
Cx43	cx43 基因具有抗癌作用		
AKT	蛋白激酶 B		

6 后记

学完这门课给我最大的感受就是我过去学的很多东西都是有用的，第一次实验就包括了：shell 脚本、python 编程、统计学、R 语言等，真正感觉到自己所学融汇到一起的感觉，尽管前面相关课程学的都很浅，但是稍加回顾就能运用的比较好，比如上课老师给的是分开的命令行，我就想编程一个自动化一点的脚本，尽管一开始连怎么读文件都忘记了，但是查阅网上资料都能很快捡起来，并且顺利编成了一个 shell 小脚本，python 也写了为数不多的函数。后面的实验包括词云，词频等我曾经接触过一些，但是当我想画自己的图来作为词云背景图的时候遇到了很多问题，但是最后为了这么一个小目标还是琢磨了好久，最后完成了这个看似没什么用的目标，到后面生物文本挖掘的过程中，通过 pubtater 可以获得很多信息，并且如果加以更深的分析就有可能有新的发现，这种发现是与我们生物学背景分不开的，在课程需要的情况下第一次在自己电脑上装上了虚拟机。在后面的 CRF、BERT 的学习中进一步了解了 NLP 的发展历程，与此同时初步接触了 pytorch 工具和神经网络，唯一的遗憾是还不能自己写代码，构建网络框架，但是掌握了如何应用这个工具，希望未来能够通过更深的学习掌握这些更加现代的工具，能够按照自己的想法搭建自己网络，并进行应用。

剩下的就是有个有趣的老师，是第一个在大学劝学生退课的老师，虽然每节课都感觉比较紧绷，甚至有时候讲算法难以跟上，但是每节课都过得很充实，并且确实有锻炼到一定的独立思考的能力和科研能力。

6.1 课程论文构思和撰写过程

当前搭建神经网络中，pytorch 是广泛使用的工具，在选题的时候想着能多接触一些近段时间的热门的技术，所以选择了 AGAC 数据库标注的任务，课上已经给出了相关的代码，只需要修改对应的参数就可以训练模型，完成 AGAC 数据库的标注任务，所以从耗时和训练的效果两个角度对两个神经网络模型进行比较，选择出合适的模型训练的 epoch。并且用 AGAC 数据库训练出来的模型去预测其他的文本，在通过预测出的实体做进一步的功能分析找到基因功能与癌症的潜在的关系。

在初始使用 pytorch 时，下载都是一大难题，虽然现在能成功运行课程代码，对于其中网络搭建看得懂一部分，并且更改一部分参数以达到自己的实验目的，但是对于内部的很多框架仍任处于不太懂的阶段。在解释 CRF 算法时，只能浅显的地方开始，即从图的角度理解 CRF 出现的原因和与 MEMM，ME 区别，在相关公式的推导方面进行了学习，虽然有些还是不能很理解，但是有了一个整体的把握，在 LSTM 和

BERT 学习上更多重心放在理解网络的具体的构建和如何使用上。通过这个课程项目，初步了解了 CRF、LSTM、BERT 算法，体会了 pytorch 在神经网络框架搭建的方便简洁，初步使用 pytorch 工具完成这次项目，BERT 模型虽然没有完美理论的支撑，但是对 NLP 领域有巨大的影响。

6.2 所参考主要资源

CRF 算法学习链接：https://www.bilibili.com/video/BV19t411R7QU?p=4&t=11
wapiti 项目链接:https://github.com/bionlp-hzau/2021Spring_CRF_AGACtask1
BERT-CRF 项目链接:https://github.com/bionlp-hzau/BERT-CRF-for-BioNLP-OST2019-AGAC-Task1
LSTM-CRF 项目链接：https://github.com/bionlp-hzau/LSTM_CRF_useAGAC

7 附录

S1. 2021 年课程-《Introduction to Conditional Random Fileds》,`https://hzaubionlp.files.wordpress.com/2020/11/5efbc89jingboslides_crfefbc8820pp4efbc89.pdf`

S2. 课程论文可以使用的基础代码，可参考 GitHub 页面, `https://github.com/bionlp-hzau/`

参考文献

[1] Y. Wang, K. Zhou, M. Gachloo, and J Xia. An overview of the active gene annotation corpus and the bionlp ost 2019 agac track tasks. In *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks*, 2019.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *arXiv*, 2017.

8.3 胡昕昀、沈敖《基于 BiLSTM 和 CRF 的序列标注》

在本次课程中，我们选择的任务是针对 AGAC 数据库进行序列标注。考虑到资源配置和运行时间等问题，我们使用了 BiLSTM 和 CRF 来完成，同时使用了 nltk 序列标注工具对 AGAC 数据库中所有的文献摘要进行分词。经过 30 个 epoch 的训练后，我们得到了初步的模型，与最开始未经训练的模型相比，模型的准确性有了显著的提高。最后，我们将 “lung cancer” 作为关键词得到 200 篇文献摘要，使用训练后的模型对其进行序列标注，试图得到一些新知识。

课程论文 GitHub 网址：`https://github.com/LikeWind99/colab`

基于 BiLSTM 和 CRF 的序列标注

胡昕昀 [1], 沈敖 [1]

[1] 华中农业大学信息学院，生物信息学，生信 1802

摘要

在本次课程中，我们选择的任务是针对 AGAC 数据库进行序列标注。考虑到资源配置和运行时间等问题，我们使用了 BiLSTM 和 CRF 来完成，同时使用了 nltk 序列标注工具对 AGAC 数据库中所有的文献摘要进行分词。经过 30 个 epoch 的训练后，我们得到了初步的模型，与最开始未经训练的模型相比，模型的准确性有了显著的提高。最后，我们将“lung cancer”作为关键词得到 200 篇文献摘要，使用训练后的模型对其进行序列标注，试图得到一些新知识。

关键词: BiLSTM、CRF、Pytorch、nltk

1 课题概况

受到刘鋆同学的倾情推荐，加上曾经在生信数学基础的课程上对夏老师的幽默风趣颇有体会，且一直以来对机器学习神经网络和自然语言处理广阔天地的兴趣使然，我们选修了这门课程。加入课程之后，发现所学内容也并非老师形容的那样如猛虎野兽般难以理解，又想体验一下神经网络的魅力，于是我们鼓起勇气挑战自我，选择了五颗星的这一课题，通过本次的课程论文也确实收获了许多。本文就在这样的情况下出生了，在这里我们计划首先训练出模型，对文本进行预测与模型评估，最后利用该模型挖掘出一些新知识。

2 数据

因为课题要求对数据库 AGAC 进行序列标注，所以我们使用 shell 从 AGAC 官网下载了其训练集、测试集以及样本集，下载地址参见附录 S1。

首先我们使用 AGAC_training 的 json 目录下所有 json 文件来训练模型，每个 json 文件的大体内容包含有该文献摘要 PubMed 网址、来源数据库、PMID（即 sourcedb）、摘要文本以及每个已标记词的 id、在摘要中所处位置及其所属的类型等信息，所属类型即为后续进行序列标注的参考。

AGAC_training 的 json 目录下包含了 250 个这样的 json 文件，我们对所有文件遍历了 30 次训练出初步的模型。另外，在 AGAC_test 的 json 目录下包含了 1000 个这样的 json 文件，作为最后测试、评估模型的数据集。AGAC_sample 则包含了 50 个训练集，用来模型预测。

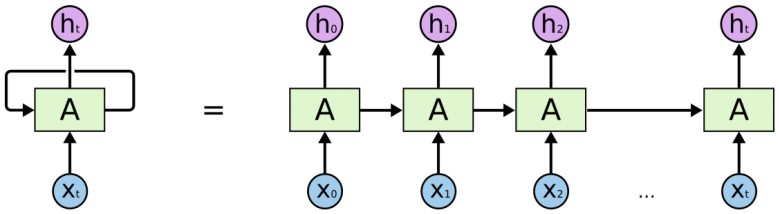
最后，我们还将该模型在肺癌相关的文献摘要上运行，数据通过 PubTator 以“lung cancer”为关键词得到，因为时间原因，只获取了 200 篇内容。

3 研究方法

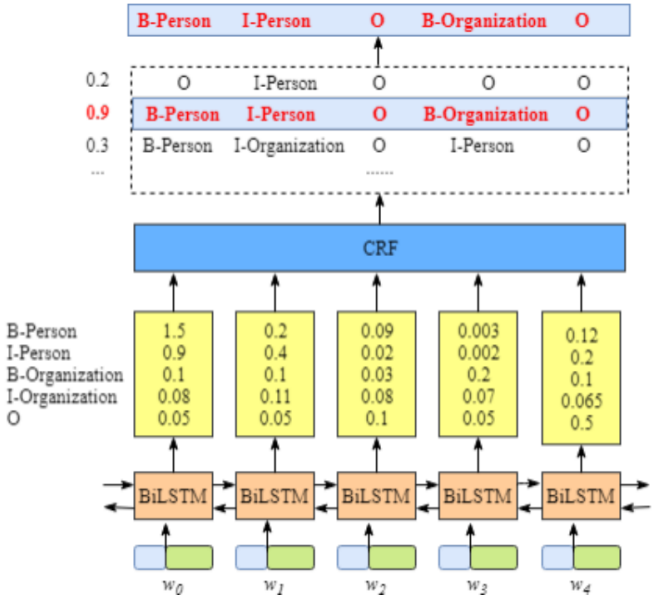
此次实验使用的是 BiLSTM 和 CRF[1]，首先我们需要了解 BiLSTM 和 CRF 的基本概念以及关键公式。

3.1 研究方法的算法背景，与其他方法的联系与区别

LSTM 是由 RNN，即循环神经网络（Recurrent Neural Network）发展而来的。RNN 的核心思想是“记忆”之前已经学习到的东西，并根据学习到的知识来进一步推断。根据图 1(a)可以看到 RNN 每次都学习一个输入 X_i，在 A 中进行一个激活操作，最后输出 h_i 并将其继续输送到下一次迭代过程中去。



(a) RNN 图示及其展开图



(b) BiLSTM-CRF 图示

图 1: RNN 与 BiLSTM-CRF 图示。图片来源：Christopher Olah[2]，课程 PPT。

但是这样会产生梯度消失或梯度爆炸的问题。因为在不断学习的过程中，最开始输入的 X_i 影响会逐渐变小，而最新的输入中如果数据十分庞大又会出现梯度爆炸。为了解决 RNN 这一弊端，将其网络结构改进的其中一种方法就是 LSTM。

利用 LSTM 对句子进行建模还存在一个问题：无法编码从后到前的信息。所以将两个方向结合在一起成为 BiLSTM，就可以更好地捕捉双向的语义依赖。

然而 BiLSTM 也存在缺陷，它在预测的时候只能预测文本序列与标签的关系，而忽略了标签之间的关联性。所以再引入 CRF，也就是条件随机场（Conditional Random Field），就可以将标签之间的转移概率，即它们之间的关系也考虑进来，随后通过最小化 LOSS 值得到最终的预测结果，如图 1(b) 所示。

3.2 研究方法中的核心思路

接下来将依次简要介绍 BiLSTM 和 CRF 的核心思想。

3.2.1 BiLSTM

BiLSTM 是双向的 LSTM。理解了 LSTM 的基本结构，BiLSTM 也是同样的原理。LSTM 包含有 cell state 和 hidden state 两个状态，并通过三个门 Forget Gate、Input Gate 和 Output Gate 来处理输入与输出。通过图 2 可以看到 LSTM 的基本结构。

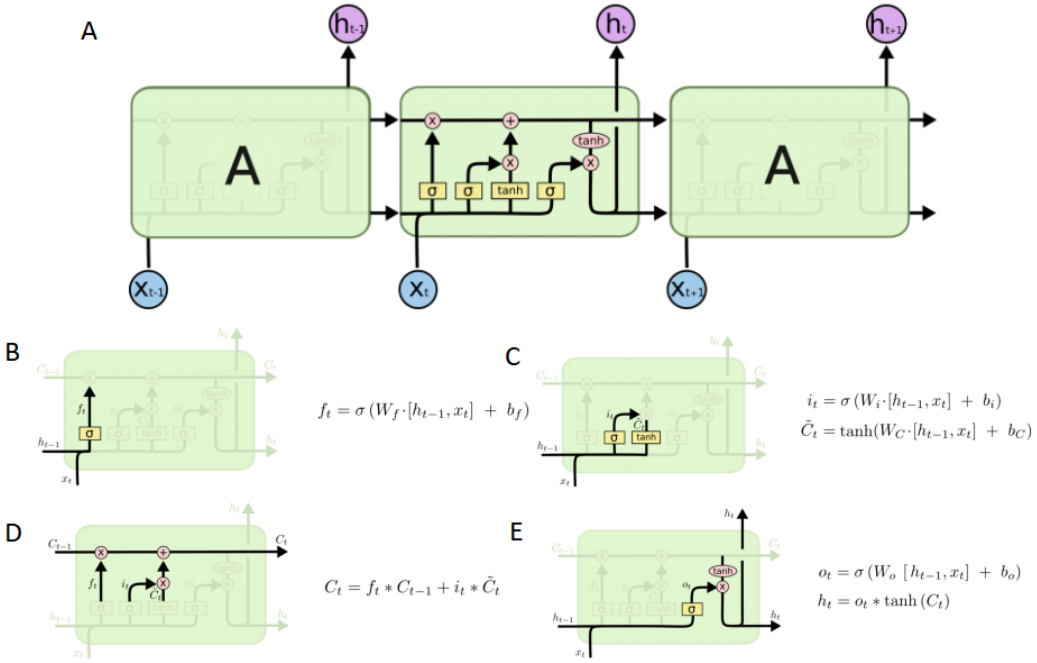


图 2: LSTM 结构图示。图片来源：Christopher Olah[2]

首先 Forget Gate 会读入上一步的输出 h_{t-1} 和当前的输入 x_t，它将决定丢弃掉哪些信息。sigmoid 函数将每个信息转化为 0 和 1 之间的数来决定其是否要丢弃。

然后决定需要存储的东西，它分为两步，首先 Input Gate 的 sigmoid 层将决定更新哪些值，然后上一步的输出 h_{t-1} 和当前的输入 x_t 将通过 tanh 层创建一个候选向量 $\tilde{C}_t$，用于后面对 cell state 的更新。

第二步将旧状态 C_{t-1} 更新为 C_t，它需要先与 f_t 相乘丢掉一些信息，然后将决定更新哪些值的门中的 $\tilde{C}_t$ 与 i_t 相乘确定需要保留的值后，二者相加就能够得到当前 cell 的存储值。

最后的 Output Gate 将决定最终需要输出的值，将当前的 C_t 通过一个 tanh 层，然后与决定输出的 o_t 相乘，一过滤就可以只输出想要输出的值。这就是一次循环的具体过程。

3.2.2 CRF

将 BiLSTM 与 CRF 结合就是由 CRF 层接受 BiLSTM 的发射分数，并得到转移分数，预测该句子向量的每个单词的类别结果。

及通过下面的公式计算出每个 $\vec{y}$ 的概率：

$$P(y|x) = \frac{1}{Z(x)} exp(\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} \mu_l s_l(y_i, x, i))$$

其中，

$$Z(x) = \sum_{y} exp(\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} \mu_l s_l(y_i, x, i))$$

式中 t_k 和 s_l 是特征函数，t_k 是定义在边上的特征函数，称为转移特征，依赖于当前和前一个位置，s_l

是定义在节点上的特征函数，称为状态特征，依赖于当前的位置，即该处的 x。其中特征函数可以简单的理解为：满足条件取 1，否则取 0 的一个函数。λ_k 和 μ_l 是对应的权值。总体来看，及分子的值是将 BiLSTM 隐藏层映射到该 $\vec{y}$ 空间的分数与标签之间的转移分数之和，而下方的值是所有的 $\vec{y}$ 分数之和。

3.3 本文的方法部分与课堂讲授内容的联系和区别/或补充

实际上整个深度学习领域在目前还属于较为“玄学”的范畴，没有科学的理论和推导，众所周知的一点是 BERT[3] 可以完全碾压 BiLSTM-CRF，但我们无法明确的说出其究竟好在哪里。也许是因为 BERT 使用了 Attention 机制，可以同时观察整个语句并给所有词汇一个概率，克服了循环神经网络一直以来想要解决的长距离依赖问题，虽然 LSTM 也有优化，但可能记忆力还是不行，看不到整个句子的特征。另外 BERT 还使用了更加优秀的 Encoder-Decoder 机制，从 Transformer 的基础上发展而来，Encoder 机制采用了短距离采样平滑，而长距离采用了跨度大的方法，并且加入了词汇的位置信息，这样可以保留更多潜在的单词间的相互关系。同时，BERT 的训练库极其庞大，每次与训练也需要花费很多时间。这些也许都让 BERT 成为了较 BiLSTM-CRF 更优秀的存在。

而 BiLSTM-CRF 相对于 BERT 的优势只是花费时间与计算资源较少，但准确性实则更为重要。

4 算法实践和代码编写要求

4.1 任务描述

我们的实验设计是首先通过 nltk 进行文本分词，使用 AGAC_training 下的文件训练模型。然后使用 AGAC_test 下的文件测试并评估模型，最后再对 AGAC_sample 目录下的文件进行预测，观察结果。BiLSTM 和 CRF 的组合是实体识别使用较广的一种方法，我们选定它也是因其适用于我们的任务，且训练所需时间相较其他算法而言稍短。

4.2 实验设计

首先我们使用 Python 的 json 库读取所有的 json 文件，使用 nltk 将其处理成以制表符分隔的每个单词、出现的文献 PMID、起始与结束位置和其所属实体标签的格式。因为原始数据集已经给出了训练集和测试集，所以我们没有对其进行更细的划分。然后就可以提取出所有的词和标注，再采用 word2idx 和 tag2idx 操作，对序列进行编码。

接着我们编辑了 BiLSTM 和 CRF 的代码，整个代码分为辅助函数和 BiLSTM_CRF 模型两部分。在 BiLSTM_CRF 模型中包含了特征提取、计算分数、前向递推、viterbi 解码以及 LOSS 计算几大模块，参考了 Pytorch 的官方文档，代码链接放置在文章附录7部分。

最后使用各个目录下的文件分别对模型进行训练、测试、评估以及预测，所有代码的运行均在 Google Colab 上完成，整体代码流程链接也放置在文章附录7部分。

4.3 某些关键代码

辅助函数中的 log_sum_exp 函数十分巧妙，它不是直接按照原式求和，而是首先找到最大值，然后将所有的分数减去这个最大值，相当于将定义域进行了平移，这样做可以避免浮点运算结果过大导致内存溢出，这种巧妙的处理方法很值得我们学习。

```
# 代码来源：https://pytorch.org/tutorials/beginner/nlp/advanced_tutorial.html?highlight=lstm
def log_sum_exp(vec):
    max_score = vec[0, argmax(vec)]
    max_score_broadcast = max_score.view(1, -1).expand(1, vec.size()[1])
```

```
    return max_score + torch.log(torch.sum(torch.exp(vec - max_score_broadcast)))
```

5 主要的生物信息学实验和实验结论

RNN 循环计算是一个十分消耗资源的过程，但为了训练出准确性更高的模型，本次实验我们进行了 30 个 epoch 的训练，花费将近 3 个小时的时间，最终训练中每个 epoch 的 LOSS 值如图 3所示。可以看到其随着迭代次数的增加呈现下降的趋势，这意味着对于训练集的准确度在逐渐增加，我们的模型相比于随机预测模型具有一定的准确性。

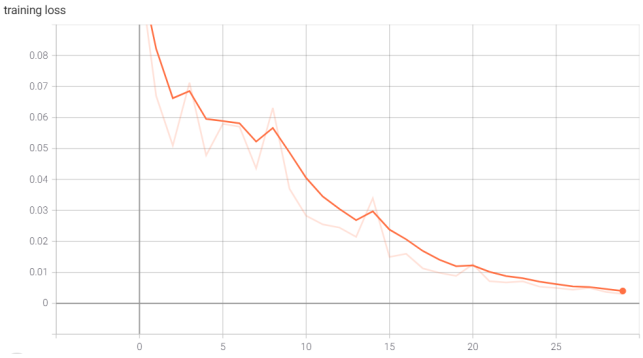


图 3: LOSS 随着迭代次数的变化。图片来源：通过 Python 的 tensorboard 库绘制。

表 1: Table example

Tag	Precision	Recall
B-Disease	0.2727272727272727	0.008955223880597015
I-Disease	0.3548387096774194	0.02689486552567237
O	0.9182434230089049	0.9992336251234715
B-NegReg	0.48717948717948717	0.052054794520547946
B-Var	0.5675675675675675	0.029453015427769985
B-Gene	1.0	0.005714285714285714
I-NegReg	1.0	0.04081632653061224
B-MPA	0.38461538461538464	0.012048192771084338
I-MPA	0.37142857142857144	0.01897810218978102
I-Gene	0.0	0.0
B-PosReg	0.5	0.003115264797507788
B-Pathway	0.0	0.0
I-Pathway	0.0	0.0
B-Protein	0.0	0.0
I-Protein	0.0	0.0
B-Interaction	0.0	0.0
B-Reg	0.0	0.0
I-Var	0.7619047619047619	0.05818181818181818
B-CPA	1.0	0.009009009009009009
I-CPA	0.3333333333333333	0.008333333333333333
I-Reg	0.0	0.0
I-PosReg	0.0	0.0
B-Enzyme	0.0	0.0
I-Enzyme	0.0	0.0
I-Interaction	0.0	0.0
Total(non-O)	0.31807354045770814	0.0509115142802196

表 1 显示的是模型评估中每个标签的精确率和回归率，结果并不算很优秀，而且有一些标签的两个值都为 0，这是什么原因呢？经过对训练文本进行查看，发现这些评估为 0 的标签出现的次数很少，也许因此模型没有得到足够的训练。同样的猜想可以类比到 O 标签，因为 O 标签出现的频率最大，所以它相应的准确率就特别高。

最后，我们还使用该模型对肺癌相关的文献摘要进行了序列标注，并观察该模型在这一领域内是否也能适用。我们使用 PubTator 通过“lung cancer”作为关键词收集了 200 篇文献摘要，随后使用该模型进行预测。可能由于 BiLSTM-CRF 本身准确率就不算特别优秀，训练次数也较少，在我们筛除了 O 标签以及一些明显预测错误的标签后，只留下了很少的单词，而且许多并不属于预测的类型，但是我们仔细观察这些筛选出来的词汇，发现它们对于研究也许仍有一定的作用，包含有治疗方法、疾病机理等方面：

1.cisplatin：顺铂。顺铂可与细胞核内 DNA 的碱基结合，造成 DNA 损伤，破坏 DNA 的复制和转录，高浓度时也抑制 RNA 及蛋白质的合成。顺铂是当前联合化疗中最常用的药物之一。

2.IMRT：IMRT 是一种先进的高精度放射线疗法，它利用计算机控制的 X 光加速器去向恶性肿瘤或肿瘤内的特定区域发射精确的辐射剂量。

3.ECM：细胞外基质，其异常表达在肿瘤浸润转移过程中发挥着重要作用。

4.T790M：T790m 的突变一般出现于非小细胞肺癌（NSCLC）的病人。如果在针对 EGFR（表皮生长因子受体家族成员之一）基因治疗之后出现了 T790m 突变，往往意味着对原来的小分子酪氨酸激酶抑制剂出现了耐药现象，这时就需要换用能够对 T790m 突变有效的新的小分子酪氨酸激酶抑制剂。

5.CD8：CD8 分子是一种白细胞分化抗原，用以辅助 T 细胞受体识别抗原并参与 T 细胞活化信号的转导。而表达 CD8 的 T 细胞 (CD8+T 细胞) 是 T 细胞里最具毒性杀伤能力的一种，可以说 CD8+T 细胞的数量直接决定了对抗肿瘤的杀伤能力。

6.MRPL15：线粒体核糖体蛋白 L15，是线粒体核糖体蛋白的一员，其异常表达与肿瘤的发生有关。MRPL15 的高表达表明 NSCLC 的预后不佳，并揭示了潜在的调节网络以及与免疫浸润的反相关。

还有一些关键词不再列出。总而言之，从这些词我们可以得到一些信息，比如肺癌可以使用顺铂等药物或 IMRT 进行治疗；与 ECM、MRPL15 的表达或脏层胸膜侵犯等有关；治疗可能导致炎症单核细胞的扩大、T790M 出现等现象；以及非小细胞肺癌（NSCLC）是肺癌中的大头，是世界上癌症相关死亡的主要原因。而利用类似 CD8+T 细胞的细胞免疫治疗前景十分广阔，被认为是最有希望攻克癌症的最佳之选。

6 后记

6.1 课程论文构思和撰写过程

本次实验的网络结构其实是相当简陋的，这样的网络的学习能力并不强，使用具有 Attention 机制的网络可能会让我们的结果更好。一开始的时候，我们就考虑使用 BERT(Bidirectional Transformers for Language Understanding)[3] 来实现整个任务，但是仔细一想 BERT 的参数数量达到了恐怖的 330M 个 (即 3.3 亿)——Google 经典的暴力美学，想要在我们现在能够的调度资源上训练这样一个庞大的网络属实有点天方夜谭了。所以为了节省计算资源、训练时间并将准确性最大化，我们最终选择了 BiLSTM-CRF 这一模型。

而且由于我们的预讲在老师教授这部分知识之前，所以不管是文本分词、将文本处理成方便使用的格式还是最终的训练测试模型，全都是我们自己编写的代码，好在沈敖同学在以前就对机器学习、深度学习颇为了解，而且网上已有大量的学习资源和代码可以参考，所以编写代码也并没有耗费过多的时间。

6.2 所参考主要资源

所有的代码均放在沈敖的 colab: `https://colab.research.google.com/drive/1pkNWZfccEtR3NZddpbEEy1_9TvAfohmj#scrollTo=zAUVaRIVWtsp`上，并且已经开通了权限，可以任意查看并运行，也可以从该项目的

GitHub: https://github.com/LikeWind99/colab上看到相关代码。参考了 Pytorch 编写 BiLSTM-CRF 的官方文档，如附录7S3 链接所示。

研究方法部分的图片和公式则来源于课程 PPT 以及附录 S2 的链接博客，结果部分图像是通过 Python 的 TensorBoard 库绘制。

6.3 代码撰写的构思和体会

总的来说，本次实验代码部分在给定参考代码的情况下其实难度并不是特别大，我们认为难度最大的地方在于如何理解 CRF 模型。在学习了 CRF 和 LSTM 的相关知识之后，再结合代码一起理解，就比单纯通过公式和概念性的东西更容易一些。而且不管是网络上的代码还是老师课程上所给的代码，都可以看到其将类封装这一点做的很好，而这也是 Python 在编写项目时十分重要且值得我们学习的。

6.4 生物信息学实验设计的构思和体会

在这里我们不仅仅局限于本次实验使用的 BiLSTM-CRF 模型来讨论。因为不管是 BiLSTM-CRF、BERT 还是其他我们还没有了解到的方法，对 NLP 的序列标注都起着很大的作用。它不仅仅可以用于这次实验中为文献里的词汇加上标签，还可以用于情感分析、信息检索、推荐和过滤等等问题。

那么在生物信息学中，我们就可以用这些方法对所有感兴趣的文献摘要进行一个初步的标注，然后再筛选出真正有用的信息。或者当收集了一些新文本时，通过这些模型预测出一些新的突变、疾病或者分子功能，从而了解该突变对物种可能产生的作用机理。因为很多时候突变和癌症可能是相互关联的，所以当预测出这些突变后，可以尝试着去搜索，查看是否与一些癌症相关联，或者对上下游的基因功能是否产生影响，也许还能发现一些前人还未研究到的内容。另外，如果是初次步入一个领域，私以为通过该方法还可以快速的了解到该领域相关的主要知识，并了解当下研究的一些热点。

比如在本次实验的最后，我们将该模型在肺癌的相关文献上运用，虽然因为训练的限制准确率还有待提升，但的确收集到了很多有用的信息，而且很快就了解到现在普遍的治疗方法以及研究进展。只不过由于准确性的问题，预测不够全面，在筛选时还需要人工助力，但是从中我们已经可以看到 NLP 与生物学相结合的无穷潜力。

6.5 人员分工

在此次任务中，沈敖负责代码编写和模型训练，胡昕昀负责论文撰写和后期模型在肺癌数据上的分析。

7 附录

S1. AGAC 数据集下载地址 http://pubannotation.org/projects/

S2. 简要理解 LSTM http://colah.github.io/posts/2015-08-Understanding-LSTMs/

S3. Pytorch 编写 BiLSTM-CRF 的官方文档 https://pytorch.org/tutorials/beginner/nlp/advanced_tutorial.html?highlight=lstm

S4. 课程论文可以使用的基础代码，可参考 GitHub 页面, https://github.com/LikeWind99/colab

参考文献

[1] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[2] Christopher Olah. Understanding lstm networks. `http://colah.github.io/posts/2015-08-Understanding-LSTMs/`, 2015.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

9

基于 Word2Vec 和 BERT 的嵌入方法探讨

是时候介绍 *Embedding* 了。

– Jingbo Xia

项目要求：

使用 Word2Vec 或者 BERT 模型进行嵌入计算。

提示：使用 PyTorch 框架，使用 Word2Vec 或者调用 BERT/Transformer 的深度语义嵌入模型，进行基本的语义嵌入计算，并通过 t-SNE 进行展示。

参考的 Word2Vec 项目链接：`https://github.com/bionlp-hzau/Tutorial_4_word2vec`

参考的 BERT 项目链接：`https://github.com/bionlp-hzau/BERT-CRF-for-BioNLP-OST2019-AGAC-Task1`

相关论文两篇：

邓启东《Word2Vec 及 BERT 模型的水稻文献词汇嵌入计算》

刘 Yun《基于 Bert 的整合水稻性状本体 (RTO) 的嵌入和展示》

9.1 邓启东《Word2Vec 及 BERT 模型的水稻文献词汇嵌入计算》

水稻由于其营养价值高，加工副产品用途广作为主要的粮食作物。截至 2020 年 6 月 2 日，PubMed 生物医学数据库中关于水稻的文献已超过 95,444 篇 [1]。人工阅读以发现其中知识固然准确，但是面对十万计的语篇数量可以说杯水车薪。因此，有关水稻文献文本挖掘方面的研究十分重要。本次实验通过 PubMed 的一个子集——6,859 篇水稻文献摘要文本作为语料库。将其中出现频率较高的前 50,000 词汇通过上下文利用经典的词嵌入方法例如 BERT，Word2Vec 训练获取其嵌入。或直接导入预训练好的模型 BERT、BioBERT、Sci-BERT 得到其嵌入。通过 t-SNE 进行降维可视化，并通过可视化结果对这几种方法进行超广比较。

课程论文 GitHub 网址：`https://github.com/LianzePuppet/article`

嵌入计算: 水稻文献词汇嵌入计算方法超广比较

邓启东 [1]

[1] 华中农业大学信息学院，430070，武汉，湖北，中国

摘要

水稻由于其营养价值高，加工副产品用途广作为主要的粮食作物。截至 2020 年 6 月 2 日，PubMed 生物医学数据库中关于水稻的文献已超过 95,444 篇 [1]。

人工阅读以发现其中知识固然准确，但是面对十万计的语篇数量可以说杯水车薪。因此，有关水稻文献文本挖掘方面的研究十分重要。本次实验通过 PubMed 的一个子集——6,859 篇水稻文献摘要文本作为语料库。将其中出现频率较高的前 50,000 词汇通过上下文利用经典的词嵌入方法例如 BERT，Word2Vec 训练获取其嵌入。或直接导入预训练好的模型 BERT、BioBERT、Sci-BERT 得到其嵌入。通过 t-SNE 进行降维可视化，并通过可视化结果对这几种方法进行超广比较。

关键词: 水稻, 词嵌入，t-SNE，Word2Vec，无监督方法

1 课题概况

选修课程的目的主要在于我们生信的培养方案中，尽管学习了 perl、R、python、shell 等多种编程语言，涉及深度学习神经网络的内容较少。而自然语言处理这门课程和生物信息学相结合，从生物文本中抽取知识，是一次学习智能算法的机会。

至于之所以选择嵌入计算作为课程论文，也在于 Word2Vec 等算法本身设计的精妙。词语的向量分布式表示和神经网络的权重矩阵不谋而合。通过梯度下降逐渐找到一套合适的嵌入满足整个上下文文本。相比于以往的各种方法而言是一大飞跃（在后文中也有所介绍）。之后的 GloVe 算法更是既通过共现矩阵统领全文，也结合上下文窗口，效果比 Word2Vec 更佳。

2 数据

实验数据详见 bionlp-hzau 教学github 项目中的 reference.table.txt，其搜集方式是通过在 Pubmed 上以 rice 作为关键词检索下载得到。

原文件表格分为六列，储存的下载的文献摘要的信息，每一列分别为标题、年份、期刊、发文机构、和基因（以 “|” 分隔）。除去标题行共计 6,859 篇摘要。

通过代码 data_preprocess.py，我们将其中的摘要抽取出来。之后将该文本做如下处理：

1. 特殊标点符号!”¤$%’()*+,./:;<=>?@[\]^_‘{|}~ 全部替换成空格
2. 将多个空格替换成一个空格
3. 将单词全部转换成小写
4. 把多个数字或者数字 + 字符 + 数字的组合替换成 NBR。

至此，我们得到了以空格分开的每个单词构成的语料库，保存在 data/corpus.txt 路径下。语料库数据准备完毕。该文件将作为后续所有方法的输入文本，使用不同方法得到嵌入，之后进行可视化。

3 研究方法

自然语言处理最基本的单位就是词语。因此想要真正了解生物文本中的知识，首先要从词语了解开始，想要了解词语，转化为嵌入是一个非常革命性的方法，因为从这时开始，词语的语义得到了很好的表示

3.1 研究方法的算法背景，与其他方法的联系与区别

3.1.1 语义分布表示前的方法

受限于语篇篇幅，这一部分的详细内容见 github 文档 “词嵌入之前的几种表示方法.doc”，链接在文末给出。详细介绍了词典、独热编码、矩阵分解等方式和它们表示词汇的优势与劣势。

3.1.2 语义分布表示

这也是我们本次嵌入计算，也就是介绍的重点内容。

语义分布表示 (distributed representation) 最早由 Hinton 提出，可以克服 one-hot representation 的上述缺点，基本思路是通过训练将每个词映射成一个固定长度的短向量，它们构成一个词向量空间，每一个向量可视为该空间上的一个点。此时向量长度可以自由选择，与词典规模无关。这是非常大的优势。

词语是自然语言处理最基本的单位。但是词语本身为符号形式，构建数学模型必须要其转化为数值型的输入。或者说——嵌入到一个数学空间里，这种嵌入方式，就叫词嵌入（word embedding) 除了 One-Hot 后面提到的几种都是此范畴。

Word2Vec

Word2Vec 是一种有效创建词嵌入的方法，它是从大量文本预料中以无监督方式学习语义知识的模型，这个模型为浅层双层的神经网络，用来训练以重新建构语言学之词文本 [3]。

Word2Vec 是轻量级的神经网络，其模型仅仅包括输入层，隐藏层和输出层，模型框架根据输入输出的不同，可以分为 CBOW 模型和 skip-gram 模型，CBOW 模型是通过上下文的内容预测中心词的可能情况，而 skip-gram 模型与其相反，它是通过中心词预测上下文词 [4]。

3.1.3 t-SNE 降维可视化

通过 t-SNE 降维可视化的方法可以通过视觉验证算法的有效性，对嵌入效果进行评估。t-SNE 是少数可以同时考虑数据全局与局部关系的算法，在很多聚类问题上的效果都不错，可以直观反映出哪些词语的距离比较近，嵌入比较相似。

3.2 研究方法中的核心思路

3.2.1 Word2Vec 的 Skip-gram 算法介绍

Skip-gram 进行上下文预测的算法重点在于不断滑动改变中心词获得上下文最终生成批处理数据的过程。在这里我们使用 “图 *1*” 进行一个详细的展示，一些文献也对 Skip-gram 中的负采样有详细的描述 [5]。

首先黑色框的部分是我们的语料库，并且由于我们已经有了一个独热编码的词语字典。因此语料库中的每一个词都可以转变为一个 index。

下面蓝色的部分是一个 buffer，它就像是一个纸带一样从整个语料库的左端向有段滑动。这里规定 skip_window（跳跃窗口）为 4，也就是每个中心词单侧的词汇量。左右两边再加上中心词自身就是一个 buffer 的大小了, 因而这里为 9。skip_num 是每次跳跃窗口选取训练模型词语，这里我们设置为 8. 实际上它一般小于两倍 skip_window，+ 我们这里取等于，意味着中心词上下的 8 个词语全部抽出。并且储存到 batch 里。

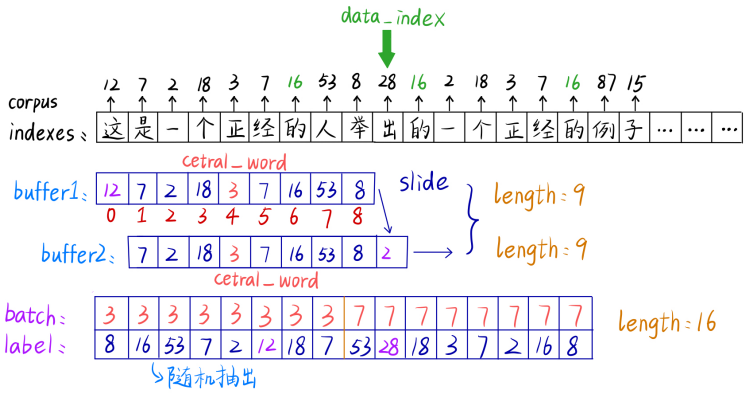


图 1: Skip-gram 算法介绍。图片来源：自绘　工具：GoodNotes 软件

batch 的 3 和 7 是中心词的 index，下面的标签是抽出的上下文词汇。我们这里绘制的 batch_size 是 16，但是实际上在代码中是 128。包括学习率在内，这些超参数均可以修改，而且会影响代码运行的时间和效果。

3.3　本文的方法部分与课堂讲授内容的联系和区别/或补充

本文其实和第七次实验的区别在于。我们拿到的语料是直接的文献文本，需要自己先对其使用 nltk 包进行预处理。处理成以空格分隔的单词作为输入。此外，本文介绍了替换预训练模型的方法，以及尝试了 GloVe、Genism 包调用 Word2Vec 等其他一些获得词嵌入的方法。

4　算法实践和代码编写要求

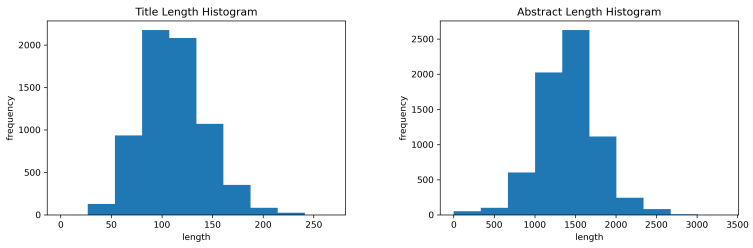
4.1　任务描述

本次实验采用六种方法或者说模型，得到同一个水稻生物医学文献摘要语料库的高频词词嵌入，通过人工检阅来直观对比各种获取词嵌入的方法模型在这一具体任务的表现情况。

4.2　实验设计

4.2.1　语料库观察分析

我们首先对语料库的 title 和 abstract 的字符数目进行一个统计（图2）。其中 (a) 和（b）分别是这 6,859 篇摘要的标题和正文的长度频率分布直方图。其实整体可以看到存在为空的，尤其是 abstract。但是最终我们都会把他们一起合并成语料库而且去掉其标点。所以去不去掉空的行对实验没有影响。



(a) 标题字符长度统计　　(b) 摘要字符长度统计

图 2: 语料库字符长度统计。图片来源：自绘　工具：Python

4.3 某些关键代码

4.3.1 绘图坐标尺度问题解决

```
代码来源：Tutorial_4_word2vec-main/Skip_Gram_basic.py第六步绘图
    x, y = low_dim_embs[i,:]
    #if x< -25 or x>35 or y < -25 or y > 25:
        #continue
    plt.scatter(x, y)
```

根据上一次的公开汇报 PPT 中所展示的图（已上传至 github：基于 PyTorch 框架的 Word2Vec 语义嵌入计算dqd.pptx)，在运行手写实现的 Word2Vec 代码的时候。如果仅仅更换作业七中的 AGAC 语料库为我们的水稻摘要语料库，并运行所给的 Skip_Gram_basic.py 代码。会发现结果就是大部分的点聚在一团看不清楚。

经过观察可以看出，存在四个词语的嵌入非常的偏离大部分词语，他们分别是 understanding、hybrid、shoot、complex，导致了坐标轴的尺度甚至达到了-70。原因分析是他们的上下文和其他词语大有不同或比较多变，还有可能词频比较低。

因此在 t-sne 将嵌入降维成二维的时候，通过限定 x,y 的范围都控制在 25 以内，如果超出 countinue 不参与绘图。以这样的方式过滤掉了比较偏离的那四个点。见图3(a),t-SNE 绘图的坐标轴变得合理了。

4.3.2 BERT、BioBERT、SciBERT 模型替换

transformers 提供用于自然语言理解（NLU）和自然语言生成（NLG）的 BERT 家族通用结构（BERT，GPT-2，RoBERTa，XLM，DistilBert，XLNet 等），包含超过 32 种、涵盖 100 多种语言的预训练模型。所有的模型都可以在 transformer 官网的预训练模型分享网站，Hugging Face[6]网站（https://huggingface.co/models）中找到。

通过替换代码 Bert_4_Rice_WordEmbedding.py 中的第 33 行配置文件中的 self.model_name 我们可以更换不同的预训练模型。将 BioBERT 模型更换为 BERT，只需要将 Bert_4_Rice_WordEmbedding.py 中代码 33 行的模型名替换即可实现调用不同的预训练模型。从未使用过的预训练模型会自动从网址下载下来。三个模型的具体模型名替换代码如下：

```
        self.model_name = 'dmis-lab/biobert-base-cased-v1.1' #BioBERT的模型名称
        self.model_name = 'bert-base-uncased'#BERT模型名
        self.model_name = 'allenai/scibert_scivocab_uncased'#SciBERT模型名
```

4.3.3 GloVe 的运行

GloVe(Global Vectors for Word Representation) 是一个基于全局词频统计的词表征工具。但是斯坦福论文 [7] 官方的代码的GitHub中可以看到，这是一个 C 的版本，相应的 python 也有网友书写，但是速度极慢。因此我们在 shell 中运行该代码（参考博客），修改 demo.sh 中的语料库为本地 corpus.txt 语料库，得到嵌入文件 vector.txt 后。还是用相同的 python 的 t-SNE 代码进行可视化，详见 Gensim_Glove 文件夹。

事实上，我们 Genism 有一个将 Glove 格式嵌入转换为 Word2Vec 的包，但是这个仅仅是词向量文件格式转换，两者区别在于 word2vec 第一行注明词向量的数量和维度。便于后续 PCA 二维可视化。但是我们是自己撰写的 t-SNE 降维可视化，因此不必在意。

(a) Word2Vec (b) Gensim-Word2Vec (c) BERT

(d) BioBERT (e) SciBERT (f) GloVe

图 3: t-SNE 可视化结果。图片来源：自绘　工具：pycharm

5　主要的生物信息学实验和实验结论

5.1　结果分析

这六张图片是我们用不同的方法获得的词嵌入，受限于版面原因，具体这六张图的高清版本可以在 github 的 figure 文件夹中获取。

5.1.1　Word2Vec

首先宏观来看，自己代码训练的 Word2Vec 聚类效果没有想象中好，整体还是呈现一团，我们仅从图中一些局部看到一些关联性。比如 root 和 seed 的距离就很近，此外还有 salinity 和 temperature 等环境因素。

猜测原因是训练的 num_steps 只有 1000，之后我们将其增加到 8000，得到新的图片，不过区别不大。也是和 Word2Vec 算法仅仅能够观察局部上下文有关，这样一来也依赖于语料库的大小。如果适当增加语料库的摘要篇数，词汇的语义相关性应该能够得到会更好的体现。

5.1.2　Genism 包中的 Word2Vec 算法

Gensim 是一款强大的开源 Python 自然语言处理工具包，里面包括很多常见模型，它支持包括 TF-IDF，LSA，LDA，和 word2vec 在内的多种主题模型算法。使用 Genism[8] 调用 Word2Vec 创建模型实际上会对数据执行两次迭代操作，第一轮操作会统计词频来构建内部的词典数结构，第二轮操作会进行神经网络训

练。详细代码见 Gensim_Word2Vec 文件夹。其中 size=100 是得到词语嵌入维度，也可以说是神经网络的层数。相比手写而言代码简单很多，训练直接使用 Word2Vec() 函数，保存得到模型使用 model.save()，保存关键嵌入为 model.wv.save_word2vec_format() 函数。之后接一个 matplotlib 画图包进行 t-SNE 可视化即可。

可以看出掉包的整体效果比我们自己写要好，初步可以看到右下角有明显的一小簇。放大可以看到 beijing、shanghai、chinese、china、hangzhou、wuhan、nanjing 这几个词语距离非常接近。而 tokyo,japan 和 korea 则在不远处。向上看可以看到和水稻疾病有关的一些词语 resistence、blast、disease、bacterial、pathogen、infection、defence 等等。中间一团稍显混乱，整体比之前好一些。

5.1.3 BERT

实现 BERT 的代码参考自作业七 AGAC 语料库的 BERT 词嵌入获取。比较关键的代码在于对 data_process.py 代码的改写，由于第七次 AGAC 语料库实验采用的格式是 Jason 格式的。其实就类似与一个字典，多了一步 jason 转变为 txt 格式的过程。我们将这部分的代码删去并改写数据读取处理。得到 rice.sentence.txt 句子文件，词频统计文件 rice.TokenFrequency.txt 和 rice.TokenFrequency.txt。详细代码见文件夹 BERT.rice。

BERT（Bidirectional Encoder Representations from Transformers）是谷歌于 2018 年发布的 NLP 领域的预训练模型，BERT 模型是使用双向 Transformer 模型的 EncoderLayer 进行特征提取。本质上是通过在海量的语料的基础上基于遮掩语言模型（Masked Language Modeling）运行自监督学习方法学习单词与单词之间的关系 [9]，由于其是一个预训练模型，已经习得一个比较好的嵌入，也就是 pre-train 阶段。之后在我们特定的水稻摘要任务中进行 fine-tune，也就是微调，预期效果应该不错。

从实际结果来看，右侧单个字母聚集在一起。左侧 gene、genome、dna、mrna、genomic、transcription 等代表基因组学的词语聚在一起。还能看到相反意义的词语 large 和 small 非常接近，相近词语 mutant 和 variety、college 和 university 及 laboratory 也很近。随意将视线专注于一个局部，会很容易发现在一起的词语存在语义关联，与预期一致。

5.1.4 BioBERT

BioBERT[10] 和 BERT 的区别在于，是一种在大型生物医学语料库上预先训练的领域特定语言表示模型。通过在任务上几乎相同的体系结构，经过生物医学语料库 (其实就是 PubMed 摘要和 PMC 全文文章) 的预训练, 见表1。

表 1: 用于 BioBERT 的文本语料库列表

语料库	单词数	所属领域
English Wikipedia	25 亿	通用
BooksCorpus	8 亿	通用
PubMed 摘要	45 亿	生物医学
PMC 全文文章	135 亿	生物医学

我们的水稻文献恰恰是来源于 PubMed 摘要，因此 BioBERT 应该更加符合我们这个特定的任务，所以预期效果应该更好。期望 BioBERT 优于 BERT 和以前的最新模型。

从图3(d) 可以明显感受到词语分成了两簇，意义相近的词语也聚得更近，和周围有一定得区分度。例如左边 germination（发芽）、seedling（幼苗）、endosperm（胚乳）、meristem（分生组织）、auxin（植物生长素）、chloroplast（叶绿体）及 chlorophyll（叶绿素）就紧紧挨在一起。左边的一大簇以名词为主，簇中也存在介词聚类，如 but、as、of、at、in、on，右边一小簇很多是一些副词动词形容词。分子生物学层面 gene、genes、protein、regulator 也靠得很近。一些词根也在一定得空间当中。从分簇得角度来说 BioBERT 效果明显好于之前几种方法。

5.1.5 SciBERT

这是在《科学文本预训练语言模型》中提出的预训练模型 [11]，它是一种接受过科学文本训练的 BERT 模型。其训练语料库是从 Semantic Scholar 上得到，训练的语料库包含 114 万文献，31 亿个分词。使用全文进行训练而不仅仅使用摘要。

从其训练结果来看，SciBERT 也呈现出自己的特点，右下角的一些介词和单个字母聚在了一起，距离很近。当一个词语只是词性或者单复数有些许区别的时候，SciBERT 也能把它们识别得很接近。可以看到很多几乎完美重合导致模糊不清的词语基本就是上述的情况。国家，地区之间也是如此，彼此的距离更加紧密。考虑到是由于语料库非常大的原因，因此经过大量统计之后，词语之间的相似性和差异得到了很好的衡量。有趣的是 amino 和 acid 几乎就在一起。tolerant、hybrid、resistant 等和水稻表型性状有关的词语倾向于得到相似的嵌入。而且可以发现 protein 和 enzyme 很近，和我们绝大多数酶都是蛋白质的知识一致。同样还有都是遗传物质的 dna 和 rna，等等。

5.1.6 GloVe 模型

GloVe 模型将奇异值分解（SVD）的 LSA 方法中全局特征的矩阵分解方法和 word2vec 局部上下文的方法结合起来。既做到了语料库全局统计（overall statistics），也通过滑动窗口得到了局部上下文特征 [7]。

最后的这个 GloVe 模型也是聚类最明显的，呈现一个“鸡腿”的形状。一些定量尺度比如 small、large、number、size、weight、length、yeild、quality 出现在最上方。ehanced、overspression、significantly、improved 等表示上调或者加强表达的词语也聚在一起。geome-wide 和 association 距离非常近，这也体现了 GloVe 能够结合局部上下文信息的特点。

6 后记

2021 年 5 月 22 日 13 时 07，中国工程院院士袁隆平老先生永远离开了我们。袁隆平老先生培育的更加优秀的籼型杂交水稻对消除全世界人民的饥饿和贫困功不可没。水稻作为我们亚洲的人口的主粮，其重要性不言而喻。

6.1 论文构思和撰写过程

由于本次论文是六种方法进行超广比较，因此主要精力在于具体工程实现，下游分析不多。在等待结果出来之余便写一写。拿到六张图片发现它们都不相同，当然同一个方法每次合成的图片也不一样。感慨就是 python 的 gensim 包来实现神经网络比自己人工书写要简单很多，在具体了解 gensim 包的使用方法之后，几句代码就可以实现模型的构建加载以及保存，在得到了嵌入之后，其实还可以进行更多如主题向量变换，文档相似计算等等更多任务的探索。也可以查询一个词汇的相似词汇。

期间主要是将实验七中的三个.py 代码弄明白，分别代表了 Word2Vec、gensim 包、BERT 三种方法。然后在原有代码基础上进行改写。得到嵌入的方法各不相同，但是可视化均采用同一个已有的 t-SNE 代码。

受制于文章篇幅，各方法的算法原理内容无法一一详述。可以留给大家自行探索，学习。

6.2 所参考主要资源

本次实验仅仅是从单词这一 NLP 的基本单元入手，以词嵌入的方式进行呈现。本次课程论文相关代码及结果获取见github 链接：https://github.com/LianzePuppet/article

参考文献

[1] E. Motschall and Y. Falck-Ytter. Searching the medline literature database through pubmed: A short guide. *Onkologie*, 28(10):517–522, 2005.

[2] D. Lin. Review of: Wordnet: An electronic lexical database. *computational linguistics*, 2002.

[3] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality arxiv : 1310 . 4546v1 [cs . cl] 16 oct 2013. 2013.

[4] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *Computer Science*, 2013.

[5] Y. Goldberg and O. Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv*, 2014.

[6] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, and M. Funtowicz. Huggingface's transformers: State-of-the-art natural language processing. 2019.

[7] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing*, 2014.

[8] Radim Rehurek and Petr Sojka. Gensim – statistical semantics in python. 2011.

[9] Bert: Pre-training of deep bidirectional transformers for language understanding. 2018.

[10] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 2019.

[11] I. Beltagy, K. Lo, and A. Cohan. Scibert: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

9.2 刘 Yun《基于 Bert 的整合水稻性状本体 (RTO) 的嵌入和展示》

生物文本中植物表型的挖掘对于作物的生物学分析、作物栽培起着至关重要的作用。植物表型存在于海量文本中，并且有多种表述形式，这给自动化挖掘带来了困难，所以一套有层次结构的标准化表述植物表型的植物性状本体的构建工作就变得十分重要。目前领域中尚不存在较为完整的水稻性状本体，于是本课程论文希望通过整合现有的描述植物性状的数据库，建立一个较为完善的水稻性状本体 (Rice Trait Ontology, RTO)，同时利用 Bert 模型对整合的性状本体 RTO 进行嵌入计算，然后利用 t-SNE 降维可视化展示嵌入结果并进行层次聚类，以检验此水稻性状本体的构建是否合理。

课程论文 GitHub 网址：`https://github.com/April-LY/Embedding-experiment-in-BioNLP-course/`

基于 Bert 的整合水稻性状本体（RTO）的嵌入计算

Liu Yun[1]

[1] 华中农业大学信息学院，430070，武汉，湖北，中国

摘要

生物文本中植物表型的挖掘对于作物的生物学分析、作物栽培起着至关重要的作用。植物表型存在于海量文本中，并且有多种表述形式，这给自动化挖掘带来了困难，所以一套有层次结构的标准化表述植物表型的植物性状本体的构建工作就变得十分重要。目前领域中尚不存在较为完整的水稻性状本体，于是本课程论文希望通过整合现有的描述植物性状的数据库，建立一个较为完善的水稻性状本体（Rice Trait Ontology, RTO），同时利用 Bert 模型对整合的性状本体 RTO 进行嵌入计算，然后利用 t-SNE 降维可视化展示嵌入结果并进行层次聚类，以检验此水稻性状本体的构建是否合理。

关键词：植物性状本体，词嵌入，Bert，t-SNE，层次聚类

1 课题概况

在大二上学期听了夏静波老师教授的《生物信息学数学基础》课后，我对数学产生了较为浓厚的兴趣，并且自主了解、学习了部分相关知识。后来又在生物物理导论课上听张红雨老师介绍《数学之美》一书，课下阅读之后对自然语言处理产生了极大的兴趣，于是想借此机会加深自己对自然语言处理方向的了解。同时单选“嵌入计算”的课程论文题目也是想借此机会巩固知识、锻炼能力。

本课程论文希望能构建一个较为完整的水稻性状本体（RTO）来帮助进行后续的生物文本水稻表型挖掘工作，同时也希望将课上所学到的生物自然语言处理相关的方法，例如 Bert [1] 模型，学以致用。本课程论文期望通过降维展示融合人本体 RTO 的词嵌入和 RTO 的层次聚类结果来说明本文构建的水稻性状本体具有一定的合理性和可用性。

2 数据

在进行本课程论文的前期调研时发现，目前领域比较认可的植物表型性状本体是 TO。最初的几个数据库如 Gramene [2] 和 Oryzabase [3] 都利用了 TO 来进行水稻表型注释。TO 是描述作物表型的受控词汇表，具有 1,554 个条目，每个条目中都包括如 **Class ID, Preferred Label, Synonyms, Definitions, Obsolete** 等的信息，如图1a 所示。同时 TO 具有清楚的层级结构（如图1b 所示）。但是 TO 的词汇有限，并且不是完全针对水稻的，所以仍需对其进行进一步的加工完善。经调查研究发现，小麦性状本体（WTO）[4] 是第一个作物性状本体，含有 596 个条目，并且也具有自己特殊的层级结构，包含的是小麦性状概念和一些环境条件（如图1c 所示）。仔细比较两个本体发现，WTO 中的环境条件并不包含在 PTO 中，且 WTO 中有许多与 PTO 中一致或者互补的植物性状。所以本论文计划将 WTO 加入 TO 以构建基础的 RTO1.0 版本。

PTO 与 WTO 支持 obo 与 csv 格式，您可以通过访问 PTO 的官方网站（`http://bioportal.bioontology.org/ontologies/PTO`）以及 WTO 的官方网站（`http://agroportal.lirmm.fr/ontologies/WHEATPHENOTYPE`）

以获取相关文件和图1b 与图1c 的动态的层次结构。图1a 展示的是 PTO 的 csv 格式的 “Plant trait” 和 “Plant height” 部分信息。

本文后续进行的本体融合工作则是利用 obo 格式文件进行操作的。本文 Bert 模型的数据集是处理过后的 RTO 词组文件。

a

Preferred Label	Synonyms	Definitions	Obsolete
Plant trait	Plant quality | Plant phenotype	A measurable or observable characteristic of a cellular component (GO:0005575), biological process (GO:0008150) or molecular function (GO:0003674) that is part of, or has participant a plant anatomical entity (PO:0025131) and/or a plant structure development stage (PO:0009012). This includes physiological, molecular and biochemical traits as well.	FALSE
plant height	PTHT (related)|shoot height (related)|Ht (related)	A whole plant morphology trait (TO:0000398) which is the height of a whole plant (PO:0000003).|Comment: Refers to: height (PATO_0000119): A 1-D extent quality inhering in a bearer by virtue of the bearer's vertical dimension of extension.	FALSE

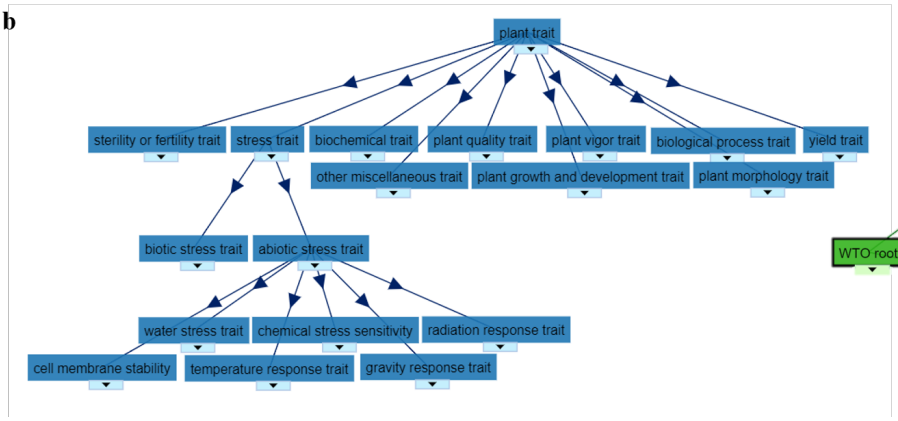


图 1: 植物形状本体（TO）条目举例及 TO 和 WTO 的层次结构展示。图片来源：a 为自己绘制，b 和 c 是来自 PTO 和 WTO 的官方网站

3 研究方法

本文的主要研究方法分为两个部分，第一个部分是人工的构建水稻性状本体（RTO），第二个部分则是利用 Bert 模型对 RTO 进行嵌入计算，同时利用生成的词嵌入进行后续的降维 t-SNE [5] 可视化展示和层次聚类。接下来的每个子章节都将从这两个部分来介绍。

3.1 人工构建水稻性状本体的背景和 Bio-Bert 模型的背景

人工构建水稻性状本体。目前已知的水稻领域的类似的工作有 funRiceGenes [6]。funRiceGenes 的工作是通过领域专家通过自身设计 526 个关键词来作为水稻性状的代表，然后对文本继续进行自动化水稻表型挖掘，这种方法虽然快速，但是有些关键词的设计如 “growth” 等很显然不是专指水稻，而这样可能会导致假阳性。而本文的本体构建工作则是依托于两个现存的可靠本体 PTO 与 WTO 进行的，通过 3 种策略（图2a）：融合、添加和删除构建了 2,022 个 RTO 条目。

Bio-Bert 模型是一个的特定领域语言表示模型，与 Bert 不尽相同，它提前在大规模生物医学语料库上预先训练，并且提供开源的代码，在生物医学 nlp 领域很大程度上优于 Bert 和以前最先进的模型。t-SNE（t-distributed stochastic neighbor embedding）：是一种非线性降维算法，适用于将高维的 Embedding 结果降维到 2 维或 3 维来降维来可视化 Embedding 之间的相似度。层次聚类：通过某种相似性测度计算节点之

间的相似性，并且将节点按照相似度从高到底排列，重新连接每个节点。最终获得一个类似于 PTO/WTO 结构的树状的层次结构网络图。

3.2 研究方法中的核心思路

人工构建水稻性状本体。我所使用的三个策略是：**删除**、**添加和融合**。对策略的操作举例展示如表1所示。

表 1: 人工构建水稻形状本体的三种策略示例

Operation	WTO term	Corresponding TO	Operation
Delete	Wheat dwarf		Delete the WTO term
Add	Grain composition	Plant quality trait	Add the WTO term to the TO father node
Merge	Nutrient deficiency	Nutrient sensitivity	Merge WTO node with its hierarchy to TO

删除：一共删除了 **63** 个节点，如图2b 所示，如果 WTO 中的本体是专属于小麦的性状，如 “wheat dwarf” 和 “wheat streak mosaic”，则我们将这个性状从 WTO 中删除，即不会被整理进入 RTO 1.0 版本中。

添加：一共添加了 **118** 个节点，如图2c 所示，依据常识和经验判断，如果 WTO 中的某个节点是属于 PTO 中某个父亲节点的子节点，则我们将这个 WTO 节点及其所有子节点一同添加到 PTO 的父亲节点下，如 “grain composition”，根据 PTO 中对 “plant quality trait” 的定义，可以确定 grain composition 是该节点的一个子节点，则此时将这个 WTO 节点添加到 PTO 的父节点下，并且仍然保留其 WTO 编号。

融合：一共融合了 **75** 个节点，如图2d 所示。当我们发现 WTO 和 PTO 中的条目是对同一性状的描述时，则我们将 WTO 这一节点下的所有节点移动到对应 PTO 下实现融合。如 TO 中的 “nutrient sensitivity” 和 “nutrient deficiency”，通常情况下这两个词都被视为同义词，于是此时我们将 WTO 本身的编号取消，同时将所有的 WTO 节点下的子节点移动到此 PTO 节点下。融合的本体以 PTO 的 TO 编号（植物性状本体的唯一 id）为主，为了避免信息丢失，将 WTO 编号记在 obo 格式的 xref（表示来自于其他本体的标签）中。

对于 Bio-Bert 模型的实施，本文所采用的代码主体部分为第七次课程作业所提供的代码，并在其基础上进行了部分改动。首先要利用 Bert 模型获得整合后的 RTO 的 Embedding，然后将 Embedding 用 t-SNE 降维展示，然后将降维结果可视化，从可视化结果中观察 RTO 中的融合与添加 WTO 条目是否与其对应的 TO 节点具有相似的嵌入。之后将 Embedding 文件利用层次聚类算法，得到 RTO 的树状层级结构，观察经算法得到的层级结构是否与我们人工整合的层级结构有相似之处。

3.3 本文的方法部分与课堂讲授内容的联系和区别/或补充

本课程论文使用的 Bio-bert 模型和课程作业 7 中的模型不同，课程中使用的是 Base 版本的，本文使用的是 large 版本，即预训练的语料库更大；同时使用的数据也不同，课程中的 data 是新冠病毒文献中出现的单个词汇种类和它们在文献中的频率，而本文的数据是水稻性状词组条目，并且没有计算频率；因为原模型是单个单词，所以不需要计算层级结构，而本文采用词组做嵌入计算并且在其本身本体就有参照层级结构的基础上，本文更进一步采用层次聚类对 Embedding 结果做了后续应用。

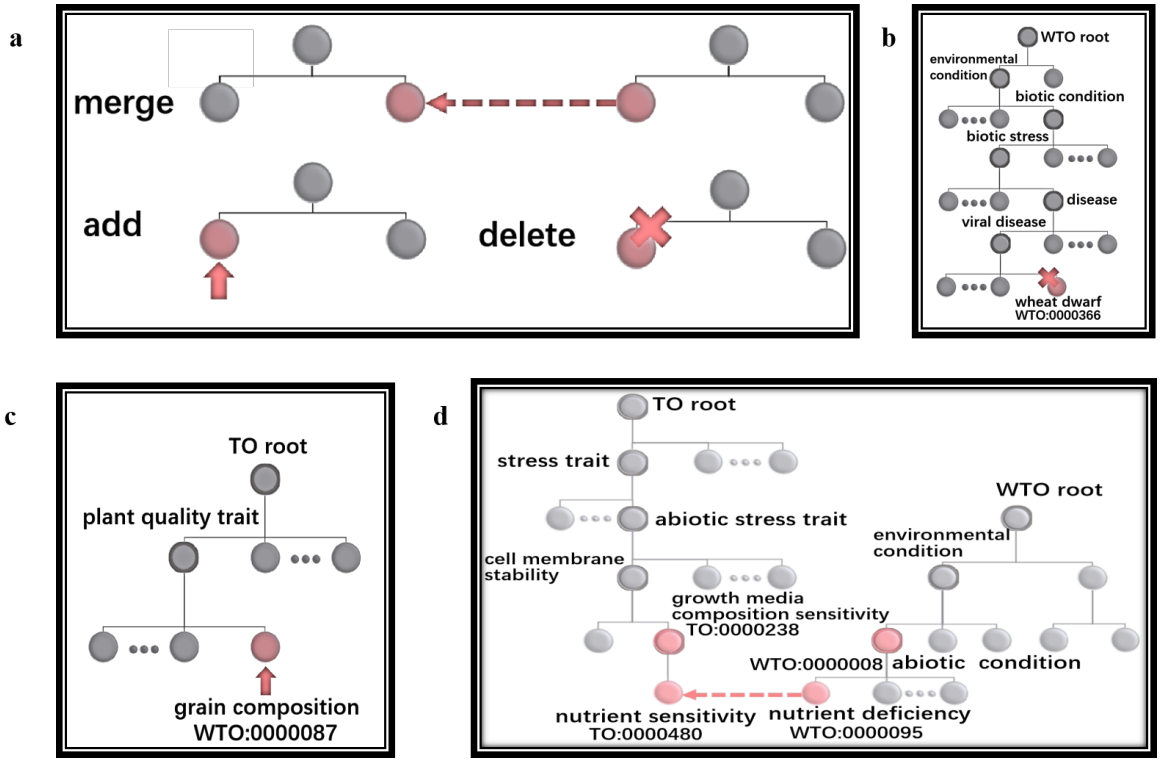


图 2: 人工构建水稻性状本体的三种策略（RTO）。图片来源：自己绘制

4 主要的生物信息学实验和实验结论

4.1 t-SNE 及层级聚类结果展示及分析

本文获得的 RTO 的 Embedding 的 t-SNE 降维可视化结果如图3c 所示，整体上分析，该图分为左右两大部分。为了更方便的展示结果，将图3c 的部分局部放大得到图3a、3b，以更好的展示结果。图3d 是根据 RTO 的 Embedding 做的层级聚类树状图。

没有语义相近的 WTO 节点聚集在原 PTO 节点附近。因为 RTO 本身就是由两个性状本体 WTO 和 PTO 整合而来，所以当观察到图 3c 中存在两个分隔比较明显的聚类时，我们可能会怀疑是不是每个聚类包括了各自的不同的性状的条目，如左边仅包括 PTO 词汇，而右边仅包括 WTO 词汇，而这种现象可能说明两个本体的条目的嵌入差异过大，本不该整合在一起。于是我们对每个聚类中的条目进行人工抽样检查，每个聚类从相隔较远的方位随机抽查，防止相似的条目聚类在一起从而导致抽样误差，如 “sensitivity” 相关的 PTO 性状全部聚集在图3b 的右侧，此时若过多抽取相关性状，则会得到大量属于 PTO 的性状。图3a 和3b 分别抽取 10 个性状如表 2 所示。不论是图3a 的聚类还是图3b 的聚类中都含有大量的 PTO 条目，几乎没有发现 WTO 条目。导致这种现象的原因可能是整合后的 PTO 条目所占比例较大，整合后的 2,022 个本体中有 1,554 个是属于原本的 PTO 的，并且部分性状的性状条目是融合（Merge）节点如”flood related trait(PTO)”，在 WTO 中对应的是 “flooding tolerance”，“anther color” 在 WTO 和 PTO 中表述相同，被我们划分为融合条目。这更减少了可视化结果中 WTO 条目出现的可能，从而无法检验 WTO 和 PTO 在语义空间上的相似性，也就无法判断我们的添加结果是否正确。

Bert 得到的嵌入结果具有一定的可靠性，可以将具有相同后缀的性状，或者层级关系较近的性状聚集在相近的语义空间上。可以看到在图 3a 中带有 “viscosity（黏性）”、“resistance”、“content”、“disease” 的前缀、后缀性状往往聚集在一起，如 “kenal smut disease resistance（玉米粒黑穗病抗性）”、“flase smut disease resistance（假黑穗病抗性）”、“smut disease”，它们因包含 “smut”、“disease” 和 “resistance”

表 2: 聚类性状本体人工抽样调查结果

聚类	调查数目	PTO 的数目	WTO 的数目
图 3a	10	pollen sterility|stigma exsertion|phenol reaction |deepwater stress|hybrid evaluation|leaf blast disease resistance |anther color|alkali digestion|radicleless|root activity	anther color
图 3b	10	fertility related trait|humidity related trait|armyworm resistance |flood related trait|seed thickness|leaf rolling time|harvest index |vascular bundle number|crown rootless|glutinous endosperm	flood related trait

拥有相近的距离，这与现实中本体的一部分情况是相符合的，现存的本体通常会将疾病（disease）和抗性（resistance）各自分为一大类，子节点则是各种各样的具体疾病和抗性。例如在图 3a 的右半部分，“chemical sensitivity” 的周围聚集了很多的 “zinc sensitivity”、“silicon sensitivity” 等条目。但是这种情况也会存在问题，如 “root color” 和 “stigma color（柱头的颜色）” 虽然在语义上很近，但是在实际上的 PTO 结构中却相隔的距离很远，因为它们会首先会被划分到不同的植物器官组织下，而不同的器官组织往往在 PTO 的层级结构中会有较远的距离，并且子条目的节点深度和类别也有不同。而这可能在后续的层次聚类中导致它们被错误的聚集到一起。

层次聚类结果不符合原本的 PTO 结构。人工检查层次聚类结果以检验其结果与 RTO 结构的一致程度。此处即选取部分例子解释层次聚类结果。如图3d 的最左边第一个绿色分支所示，其聚类包含的性状有 “Indole-3-acetic acid content” 和 “root system xx content” 以及 “shoot system xx content” 等一系列性状与原来的 PTO 结构大致上吻合，但是原本的 PTO 结构通常将 “root system calcium content” 和 “shoot system calcium content” 性状归在对应的父节点 “calcium content trait” 下面，而此层级聚类结果只是单纯的将相同前缀如 “root system” 和相同后缀的 “content” 聚集在一起。忽略掉了 PTO 中原本存在的父节点信息。此外 WTO 中的 “shoot density” 作为一个添加节点，与其最相似的是其 WTO 本身的子节点 “shoot number per plant”，这是符合显示 WTO 结构情况的。但是其周围与它相似的 PTO 节点均来自其它不同层次结构的 PTO 节点，如 “root number”、“leaf number” 等，仅仅与该节点具有相同后缀，这也与上文第二个结果的情况一致，该层次聚类结果本身已经无法正确反应 PTO 的层次结构，则更无法正确反映整合之后的 RTO 结构，以 PTO 为主的层级结构关系都不准确，则有可能原本正确添加的 RTO 结果，也因为此原因导致聚类错误。所以无法达到我们预期的人工的 RTO 结构与聚类的 RTO 结果相对比的目标。(层次聚类图片的坐标轴标签过于密集，可以于 GitHub 项目上查看高清原图，可以将标签结果复制下来查看聚类。)

5 算法实践和代码编写要求

5.1 任务描述

本代码的设计分为三个部分，预处理得到 Bert 运算的输入文件，利用 Bert 进行嵌入计算得到 RTO 条目的 Embedding，将 Embedding 结果进行 t-SNE 降维可视化展示和层级聚类。

5.2 实验设计

数据预处理：首先通过人工阅读 WTO 按照上述三条策略整理出一份本体 obo 格式文件，然后在 colab 上撰写文件预处理代码，提取出 RTO 中所有的性状词汇，即 “name：plant trait” 中的 “plant trait” 信息，得到一列 RTO 性状词汇文件，但是没有频率信息，对应课程代码中的 token 和 tokenlow 文件。之后改变课程作业 7 中的 python 脚本中的 config 信息，和 readTokenFreq 函数，来运行 Bert 计算 Embedding 结果。同时您可以在 GitHub 首页中利用 colab 打开 dataPreprocess.ipynb 文件，逐行运行其中代码即可获得

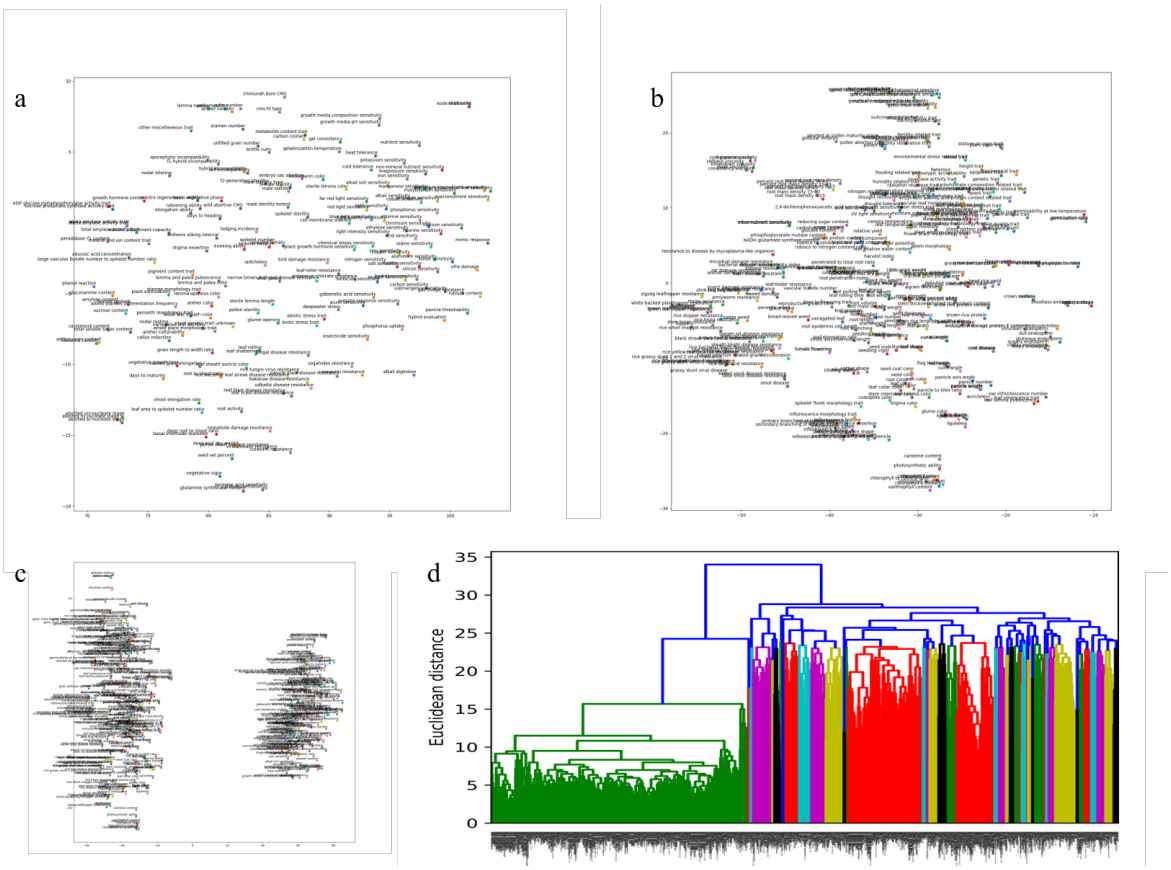


图 3: t-SNE 降维可视化展示以及层次聚类结果展示。图片来源：自己绘制

Embedding 结果。之后将得到的 Embedding 利用 python 中提供的 sklearn 包计算性状 Embedding 之间的距离，然后进行层次聚类（在下节展示关键代码）。代码在 GitHub 中的 src 目录下的 hierarchyCluster.py 中展示。

5.3 某些关键代码

```
代码来源：https://www.it610.com/article/1280291550353440768.htm
#计算嵌入之间的距离
row_dist = pd.DataFrame(squareform(pdist(df,metric='euclidean')),columns=trait,index=trait)
#依据距离进行聚类
row_clusters = linkage(pdist(df,metric="euclidean"),method='complete')
print(pd.DataFrame(row_clusters,columns=['row label1','row label2','distance','no. of items in clust.'],
    index=['cluster %d'%(i+1) for i in range(row_clusters.shape[0])]))
#层次聚类树
row_dendr = dendrogram(row_clusters,labels=trait,leaf_font_size=0.5)
```

6 后记

6.1 课程论文构思和撰写过程

有关本课程论文的构思，感谢夏静波老师和姚欣智师兄以及邓启东和沈敖同学的帮助。在与夏静波老师的讨论过程中，夏老师帮助我打开了课程论文的思路。才有了新的检验 RTO 的合理性的方法——通过层次聚类。感谢小姚师兄提供的清晰易懂的代码，和在最后一次实验课上精彩的讲授。这为我进行最后课程

论文代码的改动打下了良好的基础。感谢启东同学对图片绘制提供的建议和帮助——Bert 嵌入结果的坐标轴调整。感谢沈敖同学推荐的谷歌的 colab 平台，该平台自带了很多 python 包，还免去很多配置环境的麻烦，并且可以很方便的存储到 GitHub 项目上。大部分课程作业可以直接在 colab 上一键运行而省去很多麻烦，还具有相对较快的运算速度。

6.2 所参考主要资源

本文有关利用 Bert 模型进行嵌入计算以及 t-SNE 可视化的代码绝大部分是使用的课程作业 7 的代码（GitHub 链接：`https://github.com/bionlp-hzau/Embedding-experiment-in-BioNLP-course`）。本课程论文对该代码的 preprocess.py 部分做了相应的改动，将其中处理 Covid-19 的 json 格式文件的源代码改成了本文提取 RTO 的代码。由于课程作业 7 中并没有使用该预处理代码，我也使用了与 tutorial 类似的方式，即在 colab 上运行该预处理代码，将文件处理好然后上传至 data 文件夹下。同时如前文所提到的，为了更好的效果，改变了 config 的设置，使用的是 “BioBERT-Large v1.1 (+ PubMed 1M)” 版本的 bert，该版本使用了更多的案例进行预训练，并且还有 30k 的自定义词汇。如想使用其他版本的 bio-bert 详见`https://github.com/dmis-lab/biobert/blob/master/README.md`。本课程论文的 GitHub 链接是：`https://github.com/April-LY/Embedding-experiment-in-BioNLP-course`。使用方式与课程作业 7 没有差异，欢迎大家访问使用。后续层次聚类的代码见 src 文件夹下的 hierarchyCluster.py。

层次聚类代码的编写主要参考自博客网站:`https://www.it610.com/article/1280291550353440768.htm`，并在其基础上进行了修改。

6.3 代码撰写的构思和体会

想要实现预期的功能必须要一行一行代码的仔细看仔细分析，即使是沿用别人的代码，也需要充分的理解。才可以下手改动。对于耗时长的代码项目，必须要从早规划，否则一个模型的结果会等待很长时间才能出现，很容易导致时间安排的不合理。选择合适的平台也十分重要，比起本地运行，借助谷歌的 colab 平台可以获得更快的运行速度。

6.4 生物信息学实验设计的构思和体会

当得到 t-SNE 降维结果展示图的时候，从图片中解读出对应的生物意义是一个十分有趣的过程。可以看到有些条目聚在一起的合理性，如数字、年份、同一单词的不同形式（单数/复数），让人感叹 Bert 的效果。但是尚存在不理想的情况，如有些本该聚集在一起的却相互分散开, 如图3c 所示，分成两个大部分，而两个大部分中却又存在在真实本体中十分相近的性状。而且我们观察到的事实是，计算机处理认为的相近和人工分类的相近存在本质上的不同，计算机更多的将前后缀相同的词组分到一起，而不如人类专家考虑的详细周到。前文结果中提到 WTO 条目太少不易于从嵌入中观察到对应结果，我认为可以使用未融合前的 WTO 本体和 PTO 本体作为 Bert 的输入文件，在增加 WTO 条目的同时，更方便检查 Merge 条目的结果。我们还可以观察到一个存在的现象是尽管我们去除了 WTO 中的小麦性状词汇，PTO 中仍存在一些大麦（barley）、小麦（wheat）相关的性状词汇，后续的处理过程应该对 PTO 中的词汇也进行清洗，但是由于我们不是领域专家，我们只能处理那些很明显的带有物种的词汇，但是如果存在某些器官是某种植物特有的，那可能会存在误判、漏判的情况，所以目前的 WTO 的整理工作也需要进行进一步的复核，同时未来仍需对 PTO 进行处理。

7 附录

S1. RTO 的整合结果. https://github.com/April-LY/Embedding-experiment-in-BioNLP-course/blob/main/data/rto.obo

S2. RTO 整合策略的说明文档. https://github.com/April-LY/Embedding-experiment-in-BioNLP-course/blob/main/final2021_0529.docx

参考文献

[1] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2018.

[2] Pankaj Jaiswal, Doreen Ware, Junjian Ni, Kuan Chang, Wei Zhao, Steven Schmidt, Xiaokang Pan, Kenneth Clark, Leonid Teytelman, Samuel Cartinhour, et al. Gramene: development and integration of trait and gene ontologies for rice. *Comparative and functional genomics*, 3(2):132–136, 2002.

[3] Yukiko Yamazaki, Shingo Sakaniwa, Rie Tsuchiya, Ken-Ichi Nonomura, and Nori Kurata. Oryzabase: an integrated information resource for rice science. *Breeding Science*, 60(5):544–548, 2010.

[4] Claire Nédellec, Liliana Ibanescu, Robert Bossy, and Pierre Sourdille. Wto, an ontology for wheat traits and phenotypes in scientific publications. *Genomics & Informatics*, 18(2):e14, 2020.

[5] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[6] Yao Wen, Guangwei Li, Yiming Yu, and Yidan Ouyang. funricegenes dataset for comprehensive understanding and application of rice functional genes. *Gigascience*, (1):1–9, 2018.

10

Acknowledgement

Thank everyone who contributes to the formation of this material.
谢谢所有帮助这个教学材料成稿的人。