北京理工大學
BEIJING INSTITUTE OF TECHNOLOGY

关键词提取

汇报人：宋策、冀温瑾、何妙、叶姿逸、邹媛婷、冷晓晗

时间：2021/11/01

北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

目录

CONTENTS

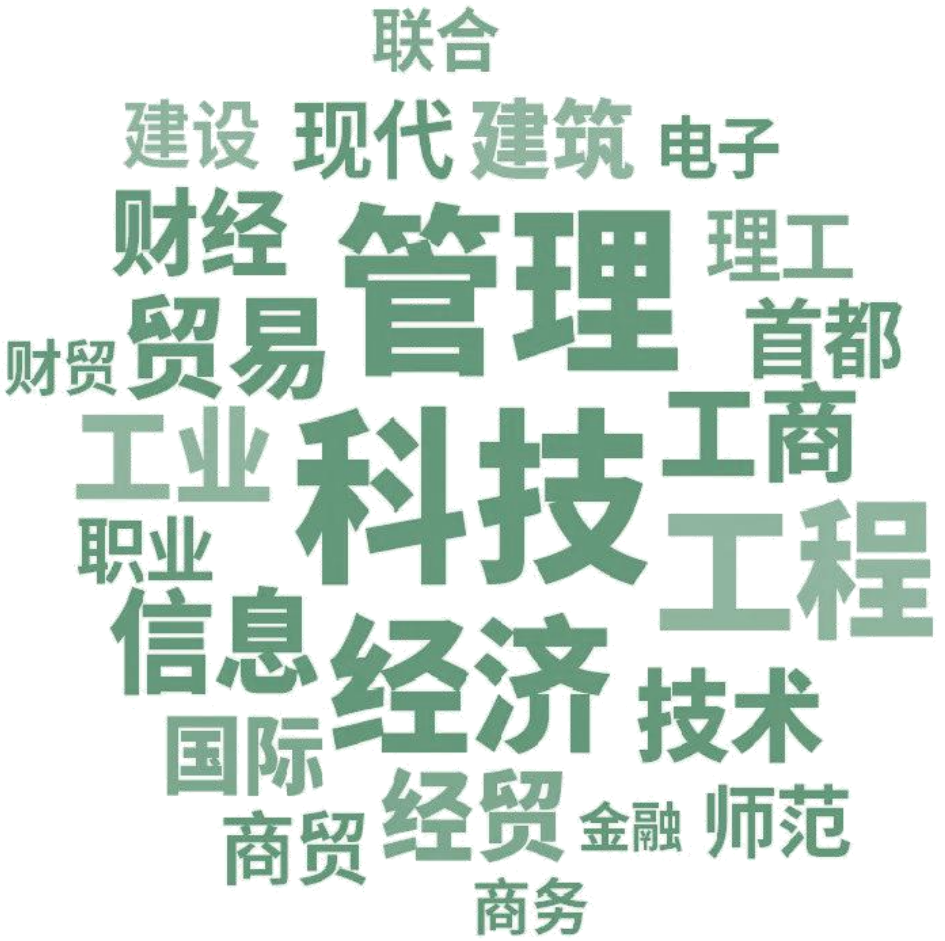
北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

1 关键词提取介绍及应用

汇报人：冀温瑾

关键词提取

关键词：是指能包含一个词素(语言中最小的有意义的单位)的词或语言里最小的可以自由运用的单位，切能够是表达文本主题内容的词，包括单词，术语和短语，在含义上是独立非复合的。包含一定的信息量，对文本内容的理解有作用。



关键词提取的应用

文献检索 由论文作者给出论文的关键词，用户可通过一个或多个关键词匹配查找到相关文献，简化搜索结果。

自动摘要 通过查找关键词最多的句子可以自动形成摘要。

推荐系统 通过用户历史浏览记录等用户标签，基于关键词匹配为用户推荐相关的产品信息。

文本分类 文本分类的核心问题是从文本中提取处关键词，然后基于一定的规则对文本分类。

搜索引擎 当我们搜索时,算法会从输入的语句中提取关键词并对文本内容进行相关性匹配。

机器翻译 机器翻译离不开关键词的抽取。

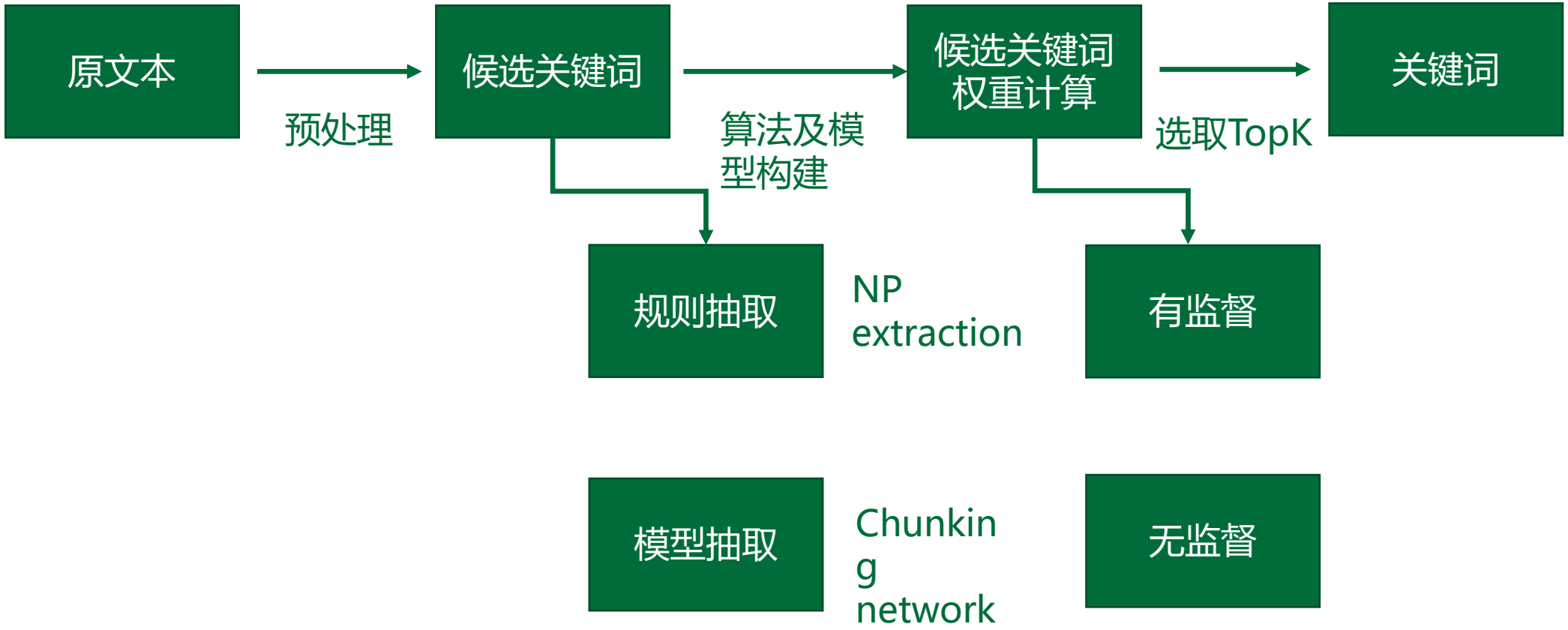
北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

2 关键词提取基本原理

汇报人：冀温瑾

关键词抽取存在最基本的问题，即什么是“关键词”，而且如何提取“关键词”。

基本流程


原文本
预处理
候选关键词
算法及模型构建
候选关键词权重计算
选取TopK
关键词
规则抽取
NP extraction
有监督
模型抽取
Chunking network
无监督

关于文本的关键词提取方法分为**有监督**和**无监督**两种：

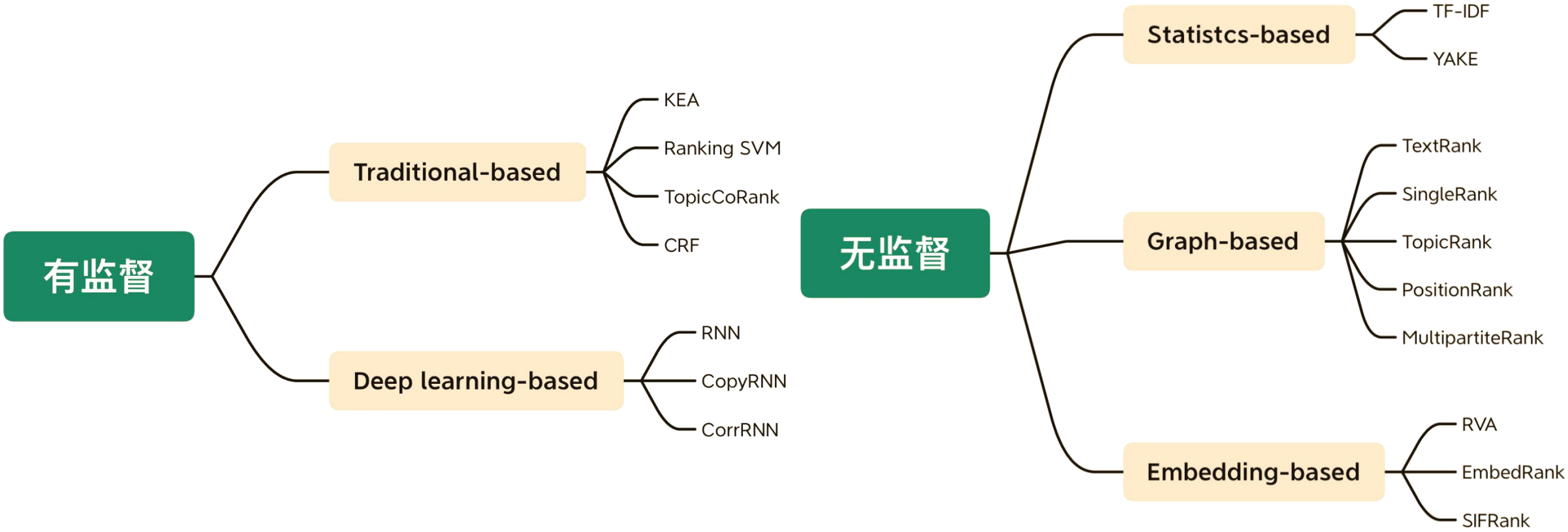
1 **有监督**

它是建关键词抽取算法看作是二分类问题，判断文档中的词或者短语是或者不是关键词。既然是分类问题，就需要提供已经标注好的训练语料，利用训练语料训练关键词提取模型，根据模型对需要抽取关键词的文档进行关键词抽取。

2 **无监督**

不需要人工标注的语料，利用某些方法发现文本中比较重要的词作为关键词，进行关键词抽取。

主要分类


有监督
Traditional-based
KEA
Ranking SVM
TopicCoRank
CRF
Deep learning-based
RNN
CopyRNN
CorrRNN
无监督
Statistcs-based
TF-IDF
YAKE
Graph-based
TextRank
SingleRank
TopicRank
PositionRank
MultipartiteRank
Embedding-based
RVA
EmbedRank
SIFRank

北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

3 经典算法介绍

汇报人：叶姿逸、何妙

TF-IDF算法

目的：用以评估一字词对于一个文件集或一个语料库中的其中一份文件的重要程度。
字词的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降。

·TF(Term Frequency)词频：词条（关键字）在文本中出现的频率

$$TF = \frac{\text{在某一类中词条}w\text{出现的次数}}{\text{该类中所有的词条数目}}$$

·IDF(Inverse Document Frequency)逆向文件频率：由总文件数目除以包含该词语的文件的数目，再将得到的商取对数得到。
·如果包含词条w的文档越少, IDF越大，则说明词条具有很好的类别区分能力。

$$IDF = log\frac{\text{语料库的文档总数}}{(\text{包含词条}w\text{的文档数}+1)}$$

(分母之所以要+1是避免分母为0)

TF-IDF算法

·某一特定文件内的高词语频率，以及该词语在整个文件集合中的低文件频率，可以产生出高权重的TF-IDF。因此，TF-IDF倾向于过滤掉常见的词语，保留重要的词语。

$$TF - IDF = TF * IDF$$

某个词对文章的重要性越高，它的TF-IDF值就越大。所以，排在最前面的几个词，就是这篇文章的关键词。

TF-IDF应用

（1）搜索引擎；（2）关键词提取；（3）文本相似性；（4）文本摘要

Text-Rank算法

TextRank算法：正规的TextRank公式在PageRank的公式的基础上，引入了边的权值的概念，代表两个句子的相似度。

TextRank 一般模型可以表示为一个有向有权图 G =(V, E), 由点集合 V和边集合 E 组成, E 是V ×V的子集。图中任两点 Vi , Vj 之间边的权重为 wji , 对于一个给定的点 Vi, In(Vi) 为 指 向 该 点 的 点 集 合 , Out(Vi) 为点 Vi 指向的点集合。点 Vi 的得分定义如下:

$$WS(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} WS(V_j)$$

其中, d 为阻尼系数, 取值范围为 0 到 1, 代表从图中某一特定点指向其他任意点的概率, 一般取值为 0.85。

Text-Rank算法

样例：

程序员(英文Programmer)是从事程序开发、维护的专业人员。一般将程序员分为程序设计人员和程序编码人员，但两者的界限并不非常清楚，特别是在中国。软件从业人员分为初级程序员、高级程序员、系统分析员和项目经理四大类。

首先对这句话分词，得出分词结果

[程序员/n, (, 英文/nz, programmer/en,), 是/v, 从事/v, 程序/n, 开发/v, 、/w, 维护/v, 的/uj, 专业/n, 人员/n, 。/w, 一般/a, 将/d, 程序员/n, 分为/v, 程序/n, 设计/vn, 人员/n, 和/c, 程序/n, 编码/n, 人员/n, ，/w, 但/c, 两者/r, 的/uj, 界限/n, 并/c, 不/d, 非常/d, 清楚/a, ，/w, 特别/d, 是/v, 在/p, 中国/ns, 。/w, 软件/n, 从业/b, 人员/n, 分为/v, 初级/b, 程序员/n, 、/w, 高级/a, 程序员/n, 、/w, 系统/n, 分析员/n, 和/c, 项目/n, 经理/n, 四/m, 大/a, 类/q, 。/w]

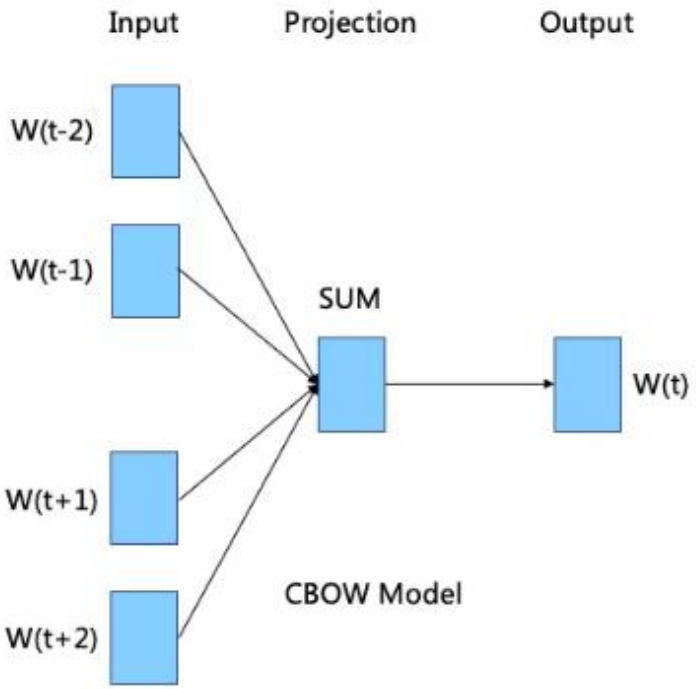
然后去掉里面的停用词，比如标点符号、常用词、以及“名词、动词、形容词、副词之外的词”。得出实际有用的词语：

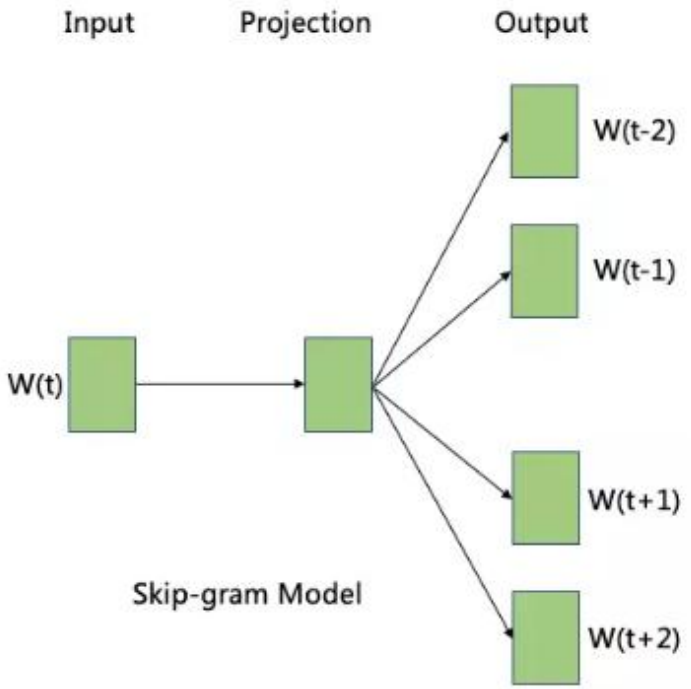
[程序员, 英文, 程序, 开发, 维护, 专业, 人员, 程序员, 分为, 程序, 设计, 人员, 程序, 编码, 人员, 界限, 特别, 中国, 软件, 人员, 分为, 程序员, 高级, 程序员, 系统, 分析员, 项目, 经理]

然后我们首先计算距离每个词距离不大于窗口大小的词的集合，然后根据公式进行迭代，直到所有词的重要性收敛到某一个值的时候，就可以停止迭代并输出结果。

word2vec算法

word2vec工具主要包含两个模型：连续词袋模型（CBOW，continuous bag of words）和跳字模型（skip-gram）





CBOW是根据上下文去预测目标词来训练得到词向量

Skip-gram是根据目标词去预测周围词来训练得到词向量

Latent Dirichlet Allocation，简称LDA，由David M. Blei、Andrew Y. Ng和Michael I. Jordan在2003年提出。

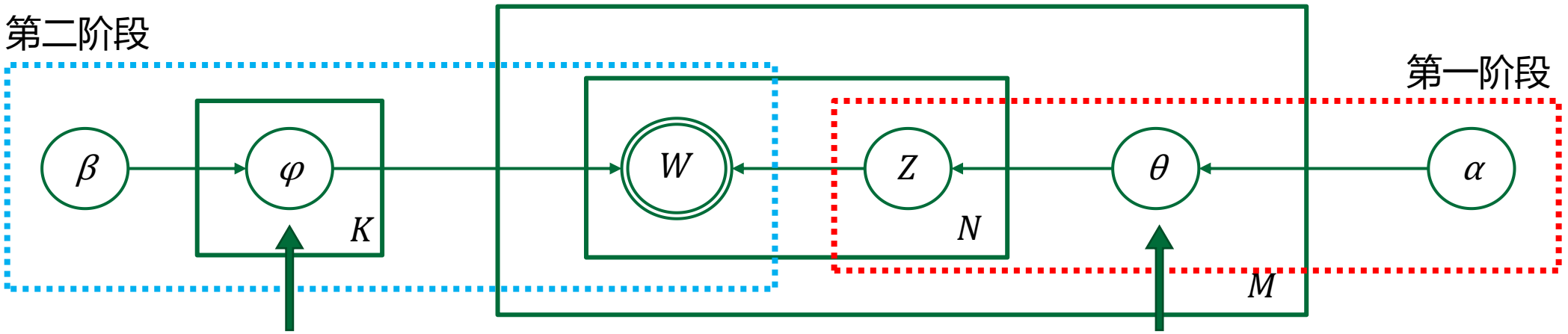
"Arts"	"Budgets"	"Children"	"Education"
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services," Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center's share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.

LDA

核心思想：①文档是若干主题的混合分布 ②每个主题又是词的概率分布

$$P(\text{词}|\text{文档}) = \sum P(\text{词}|\text{主题}) \times P(\text{主题}|\text{文档})$$

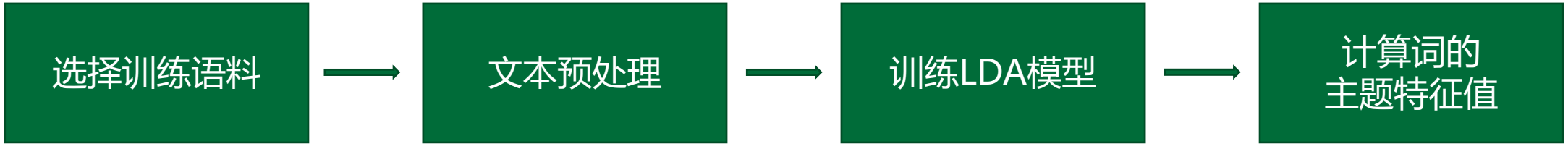


文档集总词汇数为V，一篇由n个词组成的文档，每个词的生成都服从多项分布。

不同文档之间的骰子不是同一个，每次为文档选一个主题骰子，这个过程也服从多项分布。

- **阶段一：**选取一个参数为θm的文档—主题分布，然后对第m篇文档的第n个词的主题，生成Zm,n的编号。从一个参数为α的Dirichlet分布中采样出一个多项分布θm，作为该文档在 k 个主题上的分布。
- **阶段二：**生成第m篇文档的第n个词。对该文档中的每个词，根据上步中的θm分布，采样出一个主题编号，然后根据主题—词分布对应参数为β的Dirichlet分布中采样出一个多项分布，作为该主 题下词的分布。

算法流程：



- **方法一：**假定每个词只能代表一个主题，取模型中各主题下权重高的 TOP *K* 个词作为该主题的词。
- **方法二：**假定主题区分度大的词应该是那些在某个主题下的权重高、而在其他主题中出现频率少的词语，每个词都只是代表其最能代表的那个主题.。计算出词的主题特征值之后，即可以作为关键词,也可以将词的主题特征值作为其他自动关键词抽取方法中词的一个特征使用。

信息增益：是一种基于熵的度量方式。在文本特征提取中，某个特征的信息增益是指有该特征和没有该特征时，为整个分类所能提供的信息量的差别。

信息熵

$$\text{Gain}(t_i) = \text{Entropy}(S) - \text{ExpectedEntropy}(S)$$

条件熵

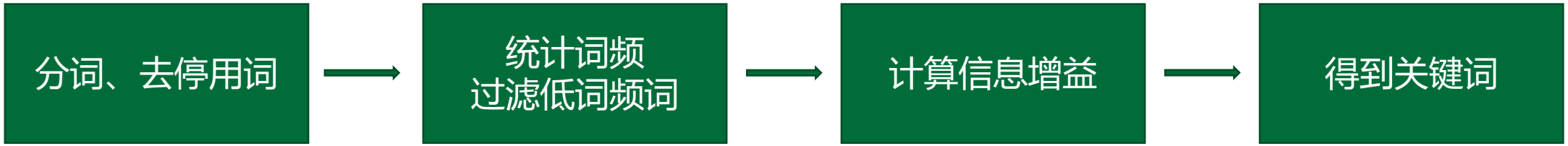
$$= \left\{ -\sum_{j=1}^{M} P(C_j) \log P(C_j) \right\}$$

$$- P(t_i) \left\{ -\sum_{j=1}^{M} P(C_j \mid t_j) \log P(C_j \mid t_j) \right\}$$

$$- P(\overline{t_1}) \left\{ -\sum_{j=1}^{M} P(C_j \mid \overline{t_1}) \log P(C_j \mid \overline{t_1}) \right\}$$

核心思想：把关键词的提取过程看成是文本分类中的特征提取的过程。在已知分类的情况下，通过将专业语料库和通用语料库进行对比，计算词条对类别信息量的贡献值，贡献值越大，其成为关键词的可能性就越大。

算法流程：



卡方检验关键词提取算法

卡方检验：卡方是数理统计中用于检验两个变量独立性的方法，是一种确定两个分类变量之间是否存在相关性的统计方法，经典的卡方检验是检验定性自变量对定性因变量的相关性。

表示 w 与 g 的共现频数

高频词g在文档中出现的期望概率

$$\chi^2(w) = \sum_{g \in G} \frac{(freq(w,g) - n_w p_g)^2}{n_w p_g}$$

词w与高频词集G中的词g共现的分布偏差

共现词的期望频率

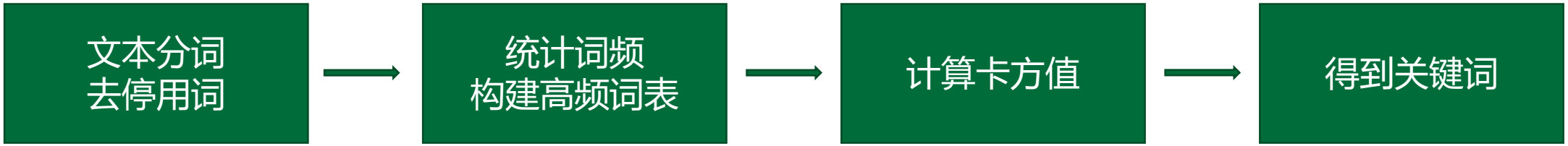
词w出现的次数

阈值 χ_α^2

卡方检验关键词提取算法

核心思想：呈特殊分布特征的词语可能是关键词，根据该规则，通过卡方检验计算词语与高频词共现的分布偏差，当偏差大于阈值时，认定该词语为关键词。

算法流程：



北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

4 开源工具与实例展示

汇报人：邹媛婷、冷晓晗、宋策

HanLP

https://github.com/hankcs/HanLP

- 面向生产环境的多语种自然语言处理工具包，基于PyTorch和TensorFlow 2.x双引擎。

- 轻量级RESTful API，仅数KB，适合敏捷开发、移动APP等场景。服务器算力有限，匿名用户配额较少，建议申请公益API秘钥auth
- 海量级native API，依赖PyTorch、TensorFlow等深度学习技术，适合专业NLP工程师、研究者以及本地海量数据场景。要求Python 3.6以上。可以在CPU上运行，推荐GPU/TPU。

主要功能：

1. 分词（粗分、细分2个标准，强制、合并、校正3种词典模式）
2. 词性标注（PKU、863、CTB、UD四套词性规范）
3. 命名实体识别（PKU、MSRA、OntoNotes三套规范）
4. 依存句法分析（SD、UD规范）
5. 成分句法分析
6. 语义依存分析（SemEval16、DM、PAS、PSD四套规范）
7. 语义角色标注
8. 词干提取
9. 词法语法特征提取
10. 抽象意义表示（AMR）
11. 指代消解
12. 语义文本相似度
13. 文本风格转换

HanLP ——关键词提取

- 内部采用**TextRank算法**实现，用户可以直接调用：
  TextRankKeyword.getKeywordList(document, size)

```
String content = "程序员(英文Programmer)是从事程序开发、维护的专业人员。一般将程序员分为程序设计人员和程序编码人员，但两者的界限并不非常清楚，特别是在中国。软件从业人员分为初级程序员、高级程序员、系统分析员和项目经理四大类。";
List<String> keywordList = HanLP.extractKeyword(content, 5);
System.out.println(keywordList);
```

```
[程序员, 程序, 分为, 人员, 软件]
```

步骤：

1. 分词
   把文本通过一个的分词算法进行分词，这里采用的是HMM算法。
2. 构造窗口
   为分个词构造窗口，这个词前后各四个词就是这个词的窗口
3. 迭代投票
   每个词最后的投票得分由这个词的窗口进行多次迭代投票决定，迭代的结束条件就是大于最大迭代次数这里是 200次，或者两轮之前某个词的权重小于某一值，这里是0.001f

关键词提取开源工具——支持中文

小明NLP

https://github.com/seanlee97/xmnlp/

- 轻量级中文自然语言处理工具
- 当前版本：2019年9月发布
- **主要功能：**
  - 中文分词 & 词性标注
  - 支持繁體
  - 支持自定义词典
  - 中文拼写检查
  - 文本摘要 & 关键词提取(**TextRank**)
  - 情感分析
  - 文本转拼音
  - 获取汉字偏旁部首

从文本中提取关键词:

xmnlp.keyword(text: str, k: int = 10, stopword: bool = True, allowPOS: Optional[List[str]] = None) -> List[Tuple[str, float]]

参数:

- text：文本输入
- k：返回关键词的个数
- stopword：是否去除停用词
- allowPOS：配置允许的词性

结果返回:

由关键词和权重组成的列表

并行处理关键词提取:

xmnlp.keyphrase_parallel(texts: List[str], k: int = 10, stopword: bool = False, n_jobs: int = 2) -> Generator[List[str], None, None]

参数:

- texts：文本列表
- n_jobs: 线程 worker 数

SnowNLP

https://github.com/isnowfy/snownlp

- 一个处理中文文本的 Python 类库

主要功能

1. 中文分词（Character-Based Generative Model）
2. 词性标注（TnT 3-gram）
3. 情感分析（现在训练数据主要是买卖东西时的评价，所以对其他的一些可能效果不是很好，待解决）
4. 文本分类（Naive Bayes）
5. 转换成拼音（Trie树实现的最大匹配）
6. 繁体转简体（Trie树实现的最大匹配）
7. 提取文本关键词（TextRank算法）
8. 提取文本摘要（TextRank算法）
9. tf-idf
10. Tokenization（分割成句子）
11. 文本相似（BM25）

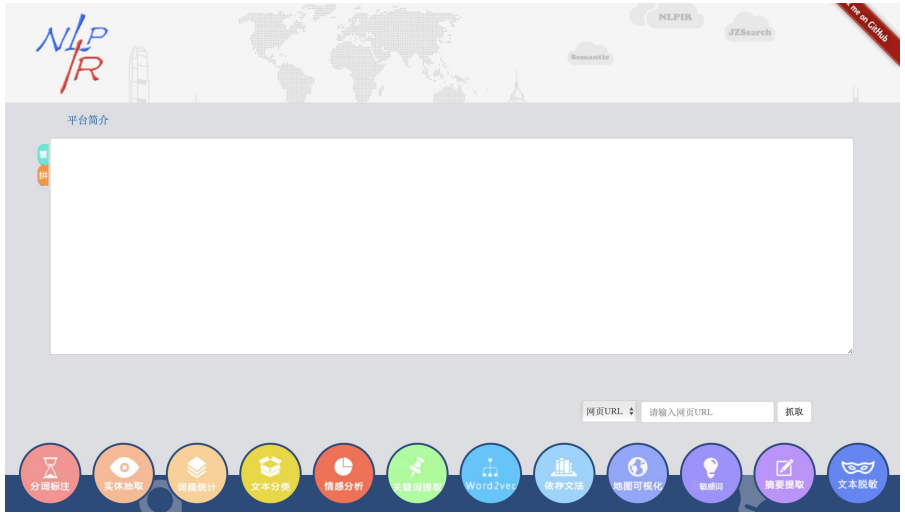
关键词提取：
keywords(self, limit=5, merge=False)

1. 输入文段
2. 按换行符和中文符号（，。？！；）划分成句子：list[句子]
3. 对每个句子分词：list[[词]]
4. 统计该词的关联词，去重并排序（关联词，该词所在句的其他词都为关联词）
5. 计算每个词的关键度（与该词的关联词的数量相关）：m[i] += 0.85 / len(list[关联词]) * m[i](old)。默认计算200次，期间前后两次m[i]相减绝对值小于等于0.001退出计算
6. 排序，截取

NLPIR-ICTCLAS汉语分词系统

https://github.com/NLPIR-team/NLPIR

- **主要功能**包括中文分词；英文分词；词性标注；命名实体识别；新词识别；关键词提取（交叉信息熵算法）；支持用户专业词典与微博分析。
- 支持多种操作系统（Windows、Linux等），多种开发语言与平台（C/C++/C#、Java、Python、Hadoop等）



- **nlpir.key_extract.import_dict(word_list: list) → list**
- **nlpir.key_extract.delete_user_word(word_list: list)**
- **nlpir.key_extract.import_blacklist(filename: str, pos_blacklist=typing.List[str]) → bool**
- **...**
- **nlpir.key_extract.get_key_words(text: str, max_key: int = 50) → List[dict]**
  - 获取文本对应的关键词,以及对应的权值,词性,词频等信息

4.1 几种支持中文开源工具对比

test_text.txt

新华社北京10月19日电　中共中央政治局10月18日下午就推动我国数字经济健康发展进行第三十四次集体学习。中共中央总书记习近平在主持学习时强调，近年来，互联网、大数据、云计算、人工智能、区块链等技术加速创新，日益融入经济社会发展各领域全过程，数字经济发展速度之快、辐射范围之广、影响程度之深前所未有，正在成为重组全球要素资源、重塑全球经济结构、改变全球竞争格局的关键力量。要站在统筹中华民族伟大复兴战略全局和世界百年未有之大变局的高度，统筹国内国际两个大局、发展安全两件大事，充分发挥海量数据和丰富应用场景优势，促进数字技术与实体经济深度融合，赋能传统产业转型升级，催生新产业新业态新模式，不断做强做优做大我国数字经济。

中国科学院院士、南京大学校长吕建教授就这个问题进行讲解，提出了工作建议。中央政治局的同志认真听取了他的讲解，并进行了讨论。

习近平在主持学习时发表了讲话。他指出，党的十八大以来，党中央高度重视发展数字经济，实施网络强国战略和国家大数据战略，拓展网络经济空间，支持基于互联网的各类创新，推动互联网、大数据、人工智能和实体经济深度融合，建设数字中国、智慧社会，推进数字产业化和产业数字化，打造具有国际竞争力的数字产业集群，我国数字经济发展较快、成就显著。特别是新冠肺炎疫情暴发以来，数字技术、数字经济在支持抗击新冠肺炎疫情、恢复生产生活方面发挥了重要作用。

习近平强调，发展数字经济是把握新一轮科技革命和产业变革新机遇的战略选择。一是数字经济健康发展有利于推动构建新发展格局，数字技术、数字经济可以推动各类资源要素快捷流动、各类市场主体加速融合，帮助市场主体重构组织模式，实现跨界发展，打破时空限制，延伸产业链条，畅通国内外经济循环。二是数字经济健康发展有利于推动建设现代化经济体系，数字经济具有高创新性、强渗透性、广覆盖性，不仅是新的经济增长点，而且是改造提升传统产业的支点，可以成为构建现代化经济体系的重要引擎。三是数字经济健康发展有利于推动构筑国家竞争新优势，当今时代，数字技术、数字经济是世界科技革命和产业变革的先机，是新一轮国际竞争重点领域，我们要抓住先机、抢占未来发展制高点。

习近平指出，要加强关键核心技术攻关，牵住自主创新这个“牛鼻子”，发挥我国社会主义制度优势、新型举国体制优势、超大规模市场优势，提高数字技术基础研发能力，打好关键核心技术攻坚战，尽快实现高水平自立自强，把发展数字经济自主权牢牢掌握在自己手中。

习近平强调，要加快新型基础设施建设，加强战略布局，加快建设高速泛在、天地一体、云网融合、智能敏捷、绿色低碳、安全可控的智能化综合性数字信息基础设施，打通经济社会发展的信息“大动脉”。要全面推进产业化、规模化应用，重点突破关键软件，推动软件产业做大做强，提升关键软件技术创新和供给能力。

习近平指出，要推动数字经济和实体经济融合发展，把握数字化、网络化、智能化方向，推动制造业、服务业、农业等产业数字化，利用互联网新技术对传统产业进行全方位、全链条的改造，提高全要素生产率，发挥数字技术对经济发展的放大、叠加、倍增作用。要推动互联网、大数据、人工智能同产业深度融合，加快培育一批“专精特新”企业和制造业单项冠军企业。要推进重点领域数字产业发展，聚焦战略前沿和制高点领域，立足重大技术突破和重大发展需求，增强产业链关键环节竞争力，完善重点产业供应链体系，加速产品和服务迭代。

习近平强调，要规范数字经济发展，坚持促进发展和监管规范两手抓、两手都要硬，在发展中规范、在规范中发展。要健全市场准入制度、公平竞争审查制度、公平竞争监管制度，建立全方位、多层次、立体化监管体系，实现事前事中事后全链条全领域监管。要纠正和规范发展过程中损害群众利益、妨碍公平竞争的行为和做法，防止平台垄断和资本无序扩张，依法查处垄断和不正当竞争行为。要保护平台从业人员和消费者合法权益。要加强税收监管和税务稽查。

习近平指出，要完善数字经济治理体系，健全法律法规和政策制度，完善体制机制，提高我国数字经济治理体系和治理能力现代化水平。要完善主管部门、监管机构职责，分工合作、相互配合。要改进提高监管技术和手段，把监管和治理贯穿创新、生产、经营、投资全过程。要明确平台企业主体责任和义务，建设行业自律机制。要开展社会监督、媒体监督、公众监督，形成监督合力。要完善国家安全制度体系。要加强数字经济发展的理论研究，就涉及数字技术和数字经济发展的问题提出对策建议。要积极参与数字经济国际合作，主动参与国际组织数字经济议题谈判，开展双多边数字治理合作，维护和完善多边数字经济治理机制，及时提出中国方案，发出中国声音。

习近平强调，数字经济事关国家发展大局，要做好我国数字经济发展顶层设计和体制机制建设，加强形势研判，抓住机遇，赢得主动。各级领导干部要提高数字经济思维能力和专业素质，增强发展数字经济本领，强化安全意识，推动数字经济更好服务和融入新发展格局。要提高全民全社会数字素养和技能，夯实我国数字经济发展社会基础。

buyuanting › 桌面 › test › tes

测试文本

人民网 >> 中国共产党新闻网 >> 习近平系列重要讲话数据库

习近平在中共中央政治局第三十四次集体学习时强调

把握数字经济发展趋势和规律 推动我国数字经济健康发展

来源：人民网-人民日报　发布时间：2021-10-20

4.1 几种支持中文开源工具对比

北京理工大學 BEIJING INSTITUTE OF TECHNOLOGY

In [1]:
```
import time
from snownlp import SnowNLP
import xmnlp
from pyhanlp import *
from nlpir import key_extract
```

In [2]:
```
with open("/[redacted]g/Desktop/test/test_text.txt", "r") as f:
    text = f.read()

s = SnowNLP(text)
t1SnowNLP = time.time()
snowNLPTags = s.keywords(5)
t2SnowNLP = time.time()
print("SnowNLP: ", end="")
print(",".join(snowNLPTags), end="")
print("")

xmnlp.set_model('/[redacted]g/Downloads/xmnlp-onnx-models')
t1xmNLP = time.time()
xmNLPTags = xmnlp.keyword(text, 5)
t2xmNLP = time.time()
print("xmnlp: ", end="")
for var in xmNLPTags:
    print(var)
print("")

t1HanLP = time.time()
HanLPTags = HanLP.extractKeyword(text, 5)
t2HanLP = time.time()
print("HanLP: ", end="")
print(",".join(HanLPTags), end="")
print("\n")

t1nlp = time.time()
nlpTags = key_extract.get_key_words(text, 5)
t2nlp = time.time()
print("NLPIR: ", end="")
for var in nlpTags:
    print(var)
```

关键词对比：

```
SnowNLP: 经济,数字,发展,产业,新

xmnlp: ('发展', 8.211491499369146)
('数字经济', 7.3708887710746)
('数字', 4.2225105950089326)
('习近平', 3.6402442137621684)
('推动', 3.329270354844655)

HanLP: 数字,经济,发展,产业,技术

NLPIR: {'freq': 29, 'pos': 'n_new', 'weight': 30.199034255113137, 'word': '数字经济'}
{'freq': 6, 'pos': 'n_new', 'weight': 21.691608909786563, 'word': '我国数字经济'}
{'freq': 31, 'pos': 'v', 'weight': 20.31373449776709, 'word': '发展'}
{'freq': 3, 'pos': 'n_new', 'weight': 14.551162767576342, 'word': '数字经济治理'}
{'freq': 7, 'pos': 'n_new', 'weight': 14.332681159996882, 'word': '数字技术'}
```

耗时对比：

```
SnowNLP: 0.4926900863647461
xmnlp: 1.3275861740112305
HanLP: 1.4350521564483643
NLPIR: 0.14505410194396973
```

YAKE

https://github.com/LIAAD/yake

- 论文：YAKE! Collection-Independent Automatic Keyword Extractor。
- 轻量级无监督自动关键字提取方法，它基于从单个文档中提取的文本统计特征来选择文本中最重要的关键字
- 不能直接作用于中文的关键词提取

关键词提取使用的特征：

- **大写term**（Casing）：大写字母的term（除了每句话的开头单词）的重要程度比那些小写字母的term重要程度要大
- **词的位置**（Word Position）：文本越开头的部分句子的重要程度比后面的句子重要程度要大
- **词频**（Term Frequency）：一个词在文本中出现的频率越大，相对来说越重要
- 上下文关系（Term Related to Context）：一个词与越多不相同的词共现，该词的重要程度越低
- **词在句子中出现的频率**（Term Different Sentence）：一个词在越多句子中出现，相对更重要

assuming default parameters

```
kw_extractor = yake.KeywordExtractor()
keywords = kw_extractor.extract_keywords(text)

for kw in keywords:
        print(kw)
```

specifying parameters

```
language = "en"
max_ngram_size = 3
deduplication_thresold = 0.9
deduplication_algo = 'seqm'
windowSize = 1
numOfKeywords = 20

custom_kw_extractor = yake.KeywordExtractor(lan=language, n=max_ngram_size, dedupLim=deduplication_t
keywords = custom_kw_extractor.extract_keywords(text)

for kw in keywords:
    print(kw)
```

- S(t)表示的是单词t的分值情况，其中s(t)分值越小，表示的单词t越重要

$$S(t) = \frac{T_{Rel} * T_{Position}}{T_{case} + \frac{TF_{norm}}{T_{Rel}} + \frac{T_{Sentence}}{T_{Rel}}}$$

SIFRank_zh

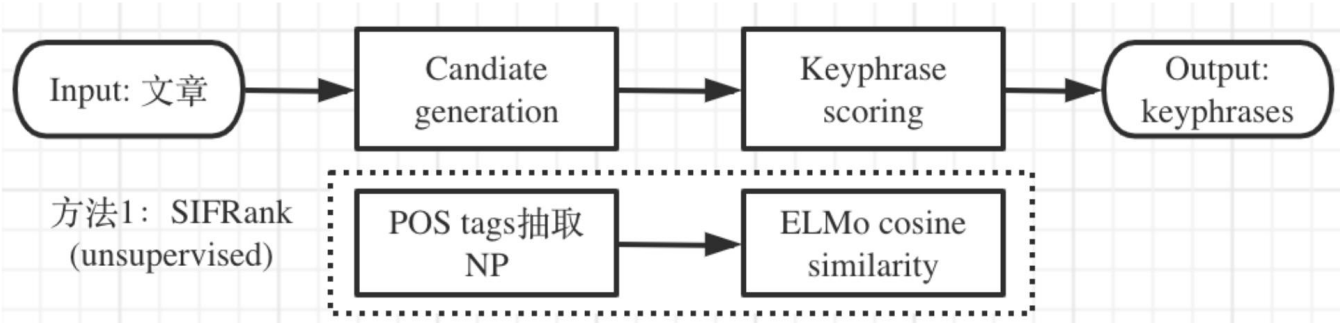
https://github.com/sunyilgdx/SIFRank_zh

- 论文：Y. Sun, H. Qiu, Y. Zheng, Z. Wang and C. Zhang, "SIFRank: A New Baseline for Unsupervised Keyphrase Extraction Based on Pre-Trained Language Model," in IEEE Access, vol. 8, pp. 10896-10906, 2020, doi: 10.1109/ACCESS.2020.2965087.
- 基于SIFRank对英文关键短语进行抽取进行改进，支持中文。
- SIFRank通过引入预训练模型，对文章和短语得到动态的sentence embedding和phrase embedding，从字、词、语法、上下文等多层次表达文章和短语，从而实现了高质量的关键短语抽取
- SIFRank++：在SIFRank的基础上融入位置信息

关键词提取：

- SIFRank利用POS tags词性标注抽取NP作为keyphrase candidates，然后利用ELMo embedding分别表示句子和短语，计算cosine similarity对候选短语进行打分。

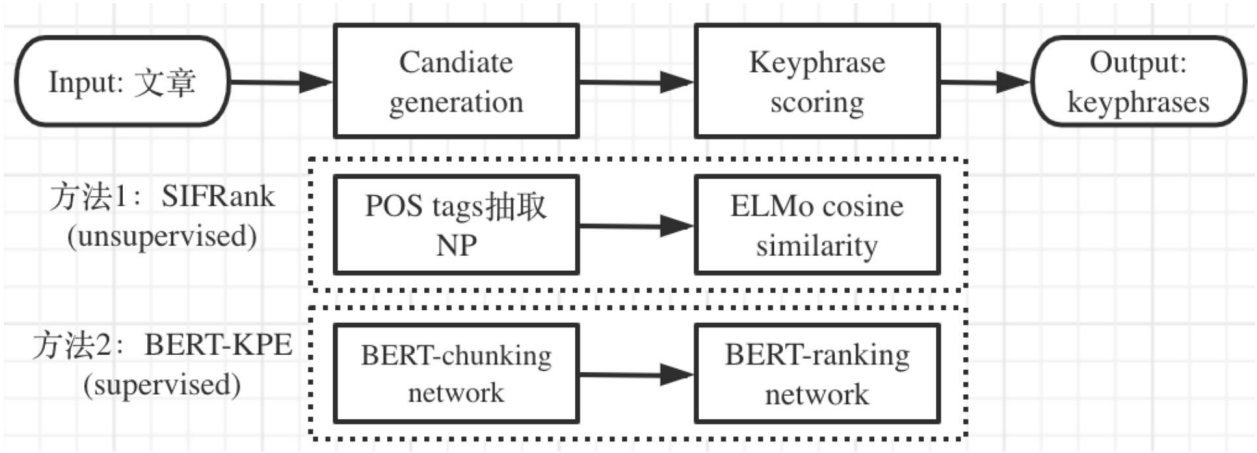
```
from embeddings import sent_emb_sif, word_emb_elmo
from model.method import SIFRank, SIFRank_plus
import thulac

#download from https://github.com/HIT-SCIR/ELMoForManyLangs
model_file = r'../auxiliary_data/zhs.model/'

ELMO = word_emb_elmo.WordEmbeddings(model_file)
SIF = sent_emb_sif.SentEmbeddings(ELMO, lamda=1.0)
#download from http://thulac.thunlp.org/
zh_model = thulac.thulac(model_path=r'../auxiliary_data/thulac.models/',user_dict=r'../auxilia
elmo_layers_weight = [0.5, 0.5, 0.0]

text = "计算机科学与技术（Computer Science and Technology）是国家一级学科，下设信息安全、软件工程、计算机
keyphrases = SIFRank(text, SIF, zh_model, N=15,elmo_layers_weight=elmo_layers_weight)
keyphrases_ = SIFRank_plus(text, SIF, zh_model, N=15, elmo_layers_weight=elmo_layers_weight)
```



BERT-KPE

https://github.com/thunlp/BERT-KPE

- 论文：Capturing Global Informativeness in Open Domain Keyphrase Extraction https://arxiv.org/abs/2004.13639.
- 使用预训练模型BERT，采用多任务学习范式，同时训练分块网络chunking network和排名网络ranking network，分别完成candidate chunking和ranking两个任务。此外，该方法也非常灵活，可以根据需要，删除任意一个子网络，仅训练另外一个子网络。



方法步骤：

1. Token embedding:采用BERT将文本编码成向量。
2. N-gram representation: 使用CNN融合窗口大小为k的字对应的token embeddings，得到相应的n-gram representation
3. Multi-task：
    1. Chunking network：预测该n-gram是否是关键短语候选
    2. Ranking network: 对n-gram进行排序

- TFIDF类的初始化

```
def __init__(self, idf_path=None):
    # 加载
    self.tokenizer = jieba.dt
    self.postokenizer = jieba.posseg.dt
    self.stop_words = self.STOP_WORDS.copy()
    self.idf_loader = IDFLoader(idf_path or DEFAULT_IDF)
    self.idf_freq, self.median_idf = self.idf_loader.get_idf()
```

jieba的TF-IDF算法抽取关键词源码分析

- 主调函数TFIDF.extract_tags

```
def extract_tags(self, sentence, topK=20, withWeight=False, allowPOS=(), withFlag=False):
    # 传入了词性限制集合
    if allowPOS:
        allowPOS = frozenset(allowPOS)
        # 调用词性标注接口
        words = self.postokenizer.cut(sentence)
    # 没有传入词性限制集合
    else:
        # 调用分词接口
        words = self.tokenizer.cut(sentence)
    freq = {}
    for w in words:
        if allowPOS:
            if w.flag not in allowPOS:
                continue
            elif not withFlag:
                w = w.word
        wc = w.word if allowPOS and withFlag else w
        # 判断词的长度是否小于2，或者词是否为停用词
        if len(wc.strip()) < 2 or wc.lower() in self.stop_words:
            continue
        # 将其添加到词频词典中，次数加1
        freq[w] = freq.get(w, 0.0) + 1.0
```

jieba的TF-IDF算法抽取关键词源码分析

- 主调函数TFIDF.extract_tags

```
# 统计词频词典中的总次数
total = sum(freq.values())
for k in freq:
    kw = k.word if allowPOS and withFlag else k
    # 计算每个词的tf-idf值
    freq[k] *= self.idf_freq.get(kw, self.median_idf) / total

# 根据tf-idf值进行排序
if withWeight:
    tags = sorted(freq.items(), key=itemgetter(1), reverse=True)
else:
    tags = sorted(freq, key=freq.__getitem__, reverse=True)
# 输出topK个词作为关键词
if topK:
    return tags[:topK]
else:
    return tags
```

jieba的TextRank算法抽取关键词源码分析

- TextRank类的初始化

```
def __init__(self):
    self.tokenizer = self.postokenizer = jieba.posseg.dt
    self.stop_words = self.STOP_WORDS.copy()
    self.pos_filt = frozenset(('ns', 'n', 'vn', 'v'))
    self.span = 5
```

jieba的TextRank算法抽取关键词源码分析

- 主调函数TextRank.textrank

```
def textrank(self, sentence, topK=20, withWeight=False, allowPOS=('ns', 'n', 'vn', 'v'), withFlag=False):

    self.pos_filt = frozenset(allowPOS)
    #定义无向有权图
    g = UndirectWeightedGraph()
    #定义共现词典
    cm = defaultdict(int)
    #分词
    words = tuple(self.tokenizer.cut(sentence))
    #一次遍历每个词
    for i, wp in enumerate(words):
        #词i满足过滤条件
        if self.pairfilter(wp):
            # 依次遍历词i 之后窗口范围内的词
            for j in xrange(i + 1, i + self.span):
                # 词j 不能超出整个句子
                if j >= len(words):
                    break
                    # 词j不满足过滤条件，则跳过
                if not self.pairfilter(words[j]):
                    continue
                    # 将词i和词j作为key，出现的次数作为value，添加到共现词典中
                if allowPOS and withFlag:
                    cm[(wp, words[j])] += 1
                else:
                    cm[(wp.word, words[j].word)] += 1
```

jieba的TextRank算法抽取关键词源码分析

- 主调函数TextRank.textrank

```
# 依次遍历共现词典的每个元素，将词i，词j作为一条边起始点和终止点，共现的次数作为边的权重
for terms, w in cm.items():
    g.addEdge(terms[0], terms[1], w)
    # 运行textrank算法
nodes_rank = g.rank()
# 根据指标值进行排序
if withWeight:
    tags = sorted(nodes_rank.items(), key=itemgetter(1), reverse=True)
else:
    tags = sorted(nodes_rank, key=nodes_rank.__getitem__, reverse=True)
# 输出topK个词作为关键词
if topK:
    return tags[:topK]
else:
    return tags
```

- UndirectWeightedGraph类的初始化及添加边函数

```
def __init__(self):
    self.graph = defaultdict(list)#这是进行分词后的一个词典

def addEdge(self, start, end, weight):
    # use a tuple (start, end, weight) instead of a Edge object
    self.graph[start].append((start, end, weight))
    self.graph[end].append((end, start, weight))
```

jieba的TextRank算法抽取关键词源码分析

- 权重迭代收敛函数rank

```
def rank(self):
    ws = defaultdict(float)#权值list表
    outSum = defaultdict(float)
    # 初始化各个结点的权值
    # 统计各个结点的出度的次数之和
    wsdef = 1.0 / (len(self.graph) or 1.0)
    for n, out in self.graph.items():
        ws[n] = wsdef
        outSum[n] = sum((e[2] for e in out), 0.0)#e[2]是什么?

    # this line for build stable iteration
    sorted_keys = sorted(self.graph.keys())
    # 遍历若干次
    for x in xrange(10):  # 10 iters
        #遍历各个节点
        for n in sorted_keys:
            s = 0
            # 遍历结点的入度结点
            for e in self.graph[n]:
                # 将这些入度结点贡献后的权值相加
                # 贡献率 = 入度结点与结点n的共现次数 / 入度结点的所有出度的次数
                s += e[2] / outSum[e[1]] * ws[e[1]]
            # 更新结点n的权值
            ws[n] = (1 - self.d) + self.d * s
```

jieba的TextRank算法抽取关键词源码分析

- 权重迭代收敛函数rank

```
(min_rank, max_rank) = (sys.float_info[0], sys.float_info[3])
# 获取权值的最大值和最小值
for w in itervalues(ws):
    if w < min_rank:
        min_rank = w
    if w > max_rank:
        max_rank = w
# 对权值进行归一化
for n, w in ws.items():
    # to unify the weights, don't *100.
    ws[n] = (w - min_rank / 10.0) / (max_rank - min_rank / 10.0)

return ws
```

爬虫与关键词提取器结合对中国新闻进行分析

无监督

分为两步：

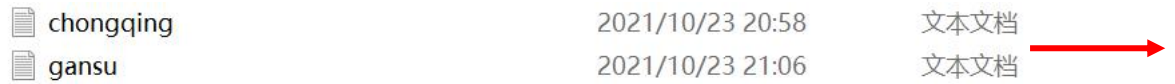
- 使用爬虫爬取相应网站的新闻信息
- 通过关键词提取器提取关键词

国内城市新闻分析：基于tfidf算法，利用jieba所提供的语料库进行计算
国际新闻分析：基于textrank

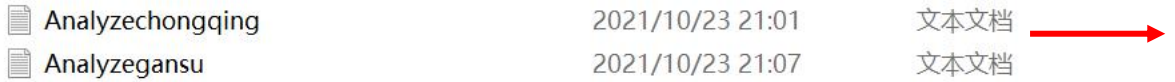
北京理工大学 BEIJING INSTITUTE OF TECHNOLOGY

爬虫与关键词提取器结合对中国新闻关注点近些年发展的分析

对于不同地区新闻的分析：
样例分别为重庆和甘肃



→ 爬虫所爬到的数据文档



→ 关键词提取器的分析结果

爬虫与关

对于不同地
样例分别为

chongqing - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

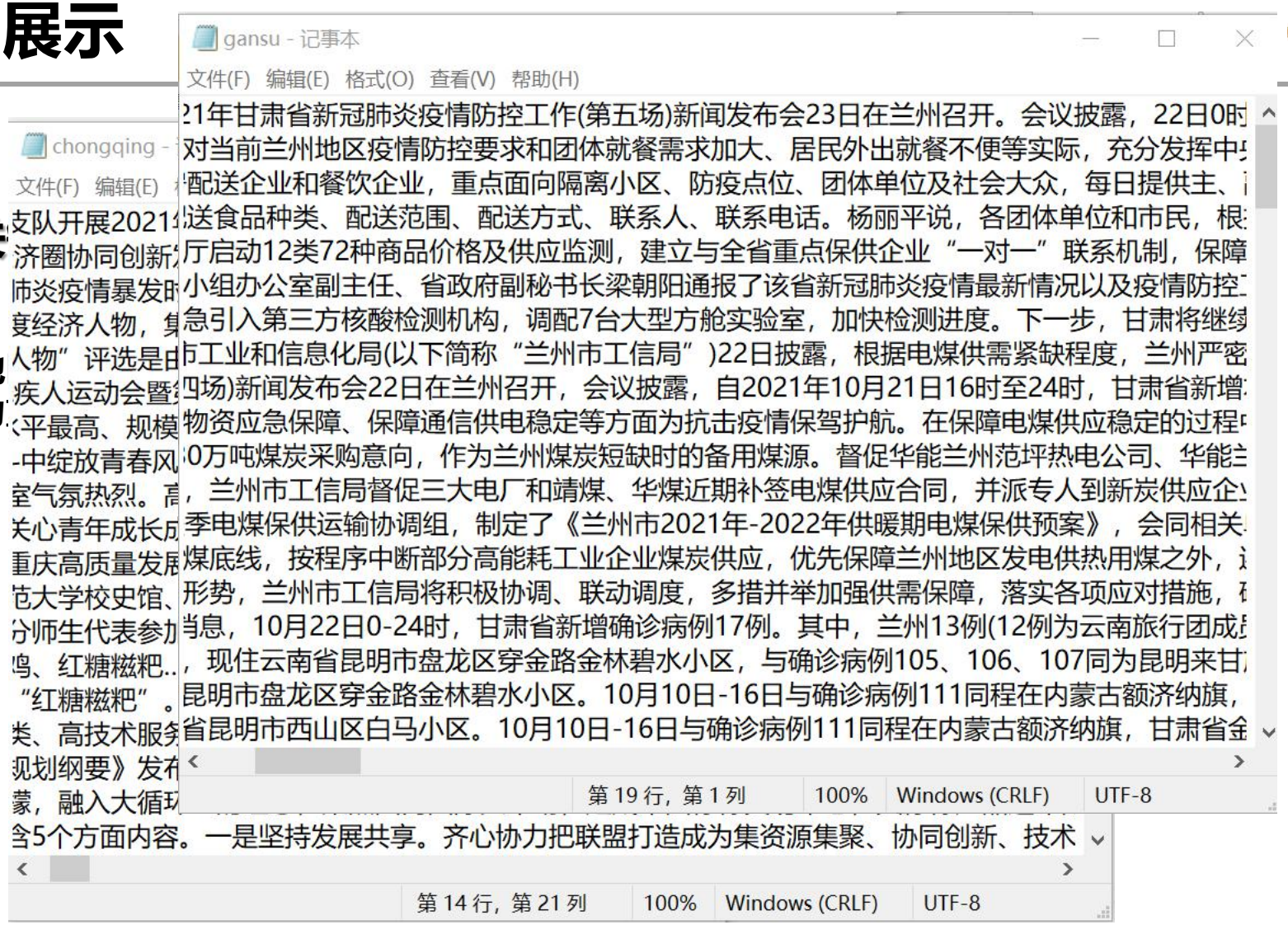
支队开展2021年度军事训练大比武暨军事体育运动会，200多名参赛队员在竞技赛场展开激烈
济圈协同创新发展峰会”同步举行了“2020十大重庆经济年度人物、十大重庆经济年度创新
肺炎疫情暴发时带领公司全体员工迎难而上成功研发核酸提取试剂和仪器的中元汇吉生物技术
度经济人物，集中展现了2020年这一年的非凡历程，具有鲜明年度特征，优中选优，是行业
人物”评选是由市委组织部、市委宣传部、市委统战部指导，重庆日报报业集团和重庆两江新
疾人运动会暨第八届特殊奥林匹克运动会(以下简称“残特奥会”)开幕式在陕西西安举行。
平最高、规模最大的综合性残疾人体育盛会。此次重庆代表团总规模为291人，其中，运动
-中绽放青春风采。10月22日，重庆市市委书记陈敏尔前往重庆大学、重庆师范大学调研座谈
室气氛热烈。高校师生代表围绕学习习近平总书记“七一”重要讲话精神发言。陈敏尔同大家
关心青年成长成才，在“七一”重要讲话中对广大青年提出殷切期望。未来属于青年，希望
重庆高质量发展提供人才和智力支撑。要切实抓好高校党建工作，加强党对高校工作的领导，
范大学校史馆、重庆大学分析测试中心调研，并听取全市教育系统开展党史学习教育情况汇报
分师生代表参加。10月19日，草坪、花、灌木等将两江新区人和立交装扮得如同“城市彩带”
鸣、红糖糍粑……10月20日，永川区朱沱镇南华宫小学的操场上，上百名学生变身“大厨”，
“红糖糍粑”。10月21日，西部(重庆)科学城科技创新项目集中签约活动举行。市委副书记、
类、高技术服务类、大健康类科技创新项目11个，总投资106亿元，包括重庆先进病理研究院
规划纲要》发布之际，10月21日，2021中国·重庆潼南国际柠檬节在潼南区国家现代农业产业
蒙，融入大循环”的理念，聚焦国内国际大市场，建成中国柠檬交易中心，其柠檬产品遍布国
含5个方面内容。一是坚持发展共享。齐心协力把联盟打造成为集资源集聚、协同创新、技术

第 14 行，第 21 列 100% Windows (CRLF) UTF-8

4.3 实例展示

爬虫与关

对于不同地
样例分别为



爬虫与关键词提取器结合对中国新闻关注点近些年发展的分析

经济发展、生态保护

疫情反复

爬虫与关键词提取器结合对中国新闻关注点近些年发展的分析

对中国新闻网国际模块的新闻分析：

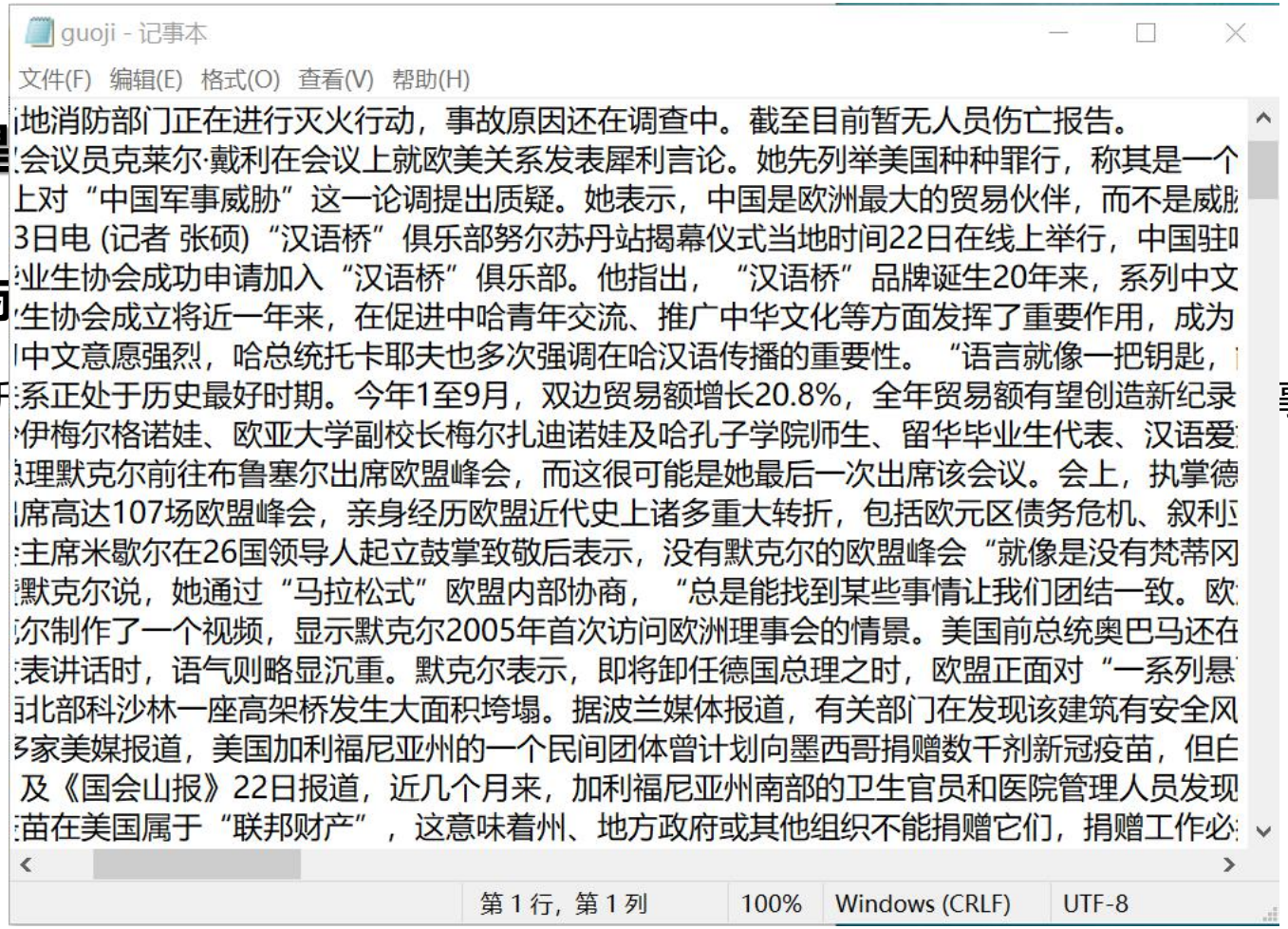
由于对该模块的分析主要是想研究近期与中国冲突较多的国家，与国际近期发生频率较高的事件。采用textrank算法。

北京理工大学 BEIJING INSTITUTE OF TECHNOLOGY

爬虫与关键词提

对中国新闻网国际

由于对该模块的分析 事件。采用textrank算法。



爬虫与关键词提取器结合对中国新闻关注点近些年发展的分析

对中国新闻网国际模块的新闻分析：

关键词分析所得到的结果：

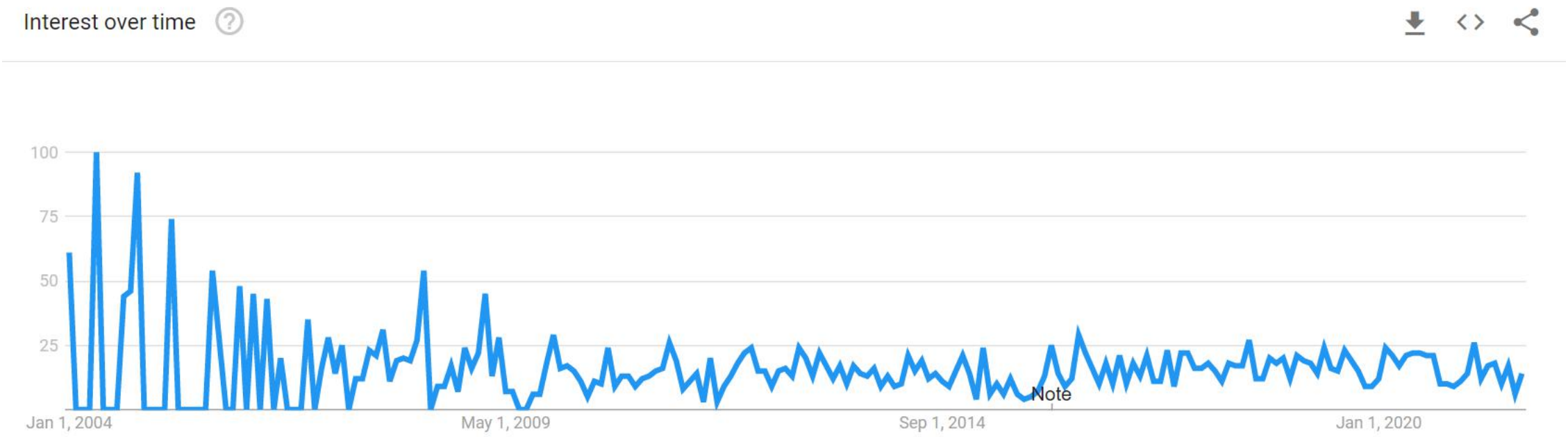
```
keywords by textRank:
美国/新冠/疫苗/病例/问题/新疆/疫情/供应链/表示/中国/
```

北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

5 关键词提取前沿研究介绍

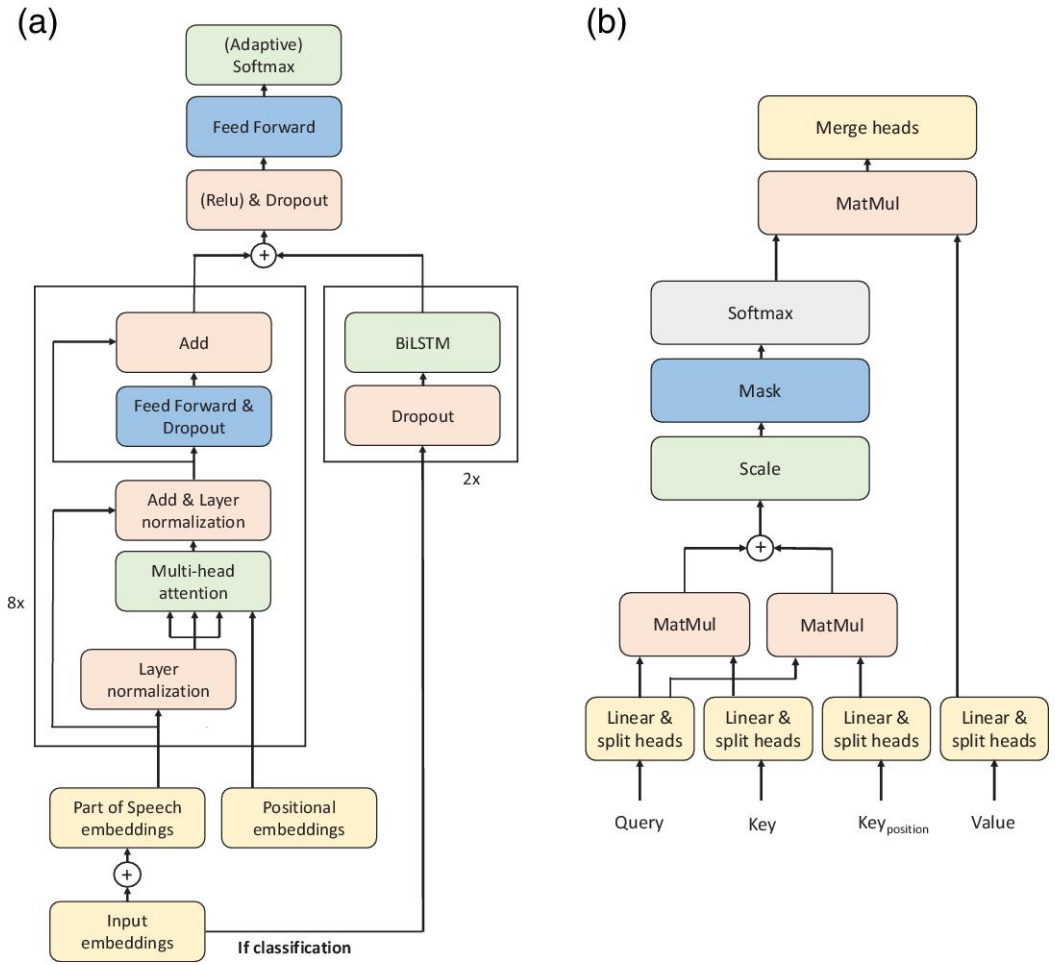
汇报人：冀温瑾

5 关键词抽取领域文献发表趋势



- Data were collected from Google Scholar by searching ‘keyword extraction’ and ‘keyphrase extraction’.

TNT-KID[1]



[1] Martinc M , Krlj B , Pollak S . TNT-KID: Transformer-based neural tagger for keyword identification[J]. Natural Language Engineering, 2021:1-40.

5 TNT-KID：实验结果

	KP20k	Inspec	Krapivin	NUS	SemEval	KPTimes	JPTimes	DUC	Average
Unsupervised algorithms									
TfIdf									
F1@5	0.072	0.160	0.067	0.112	0.088	0.179*	0.266*	0.098*	0.130
F1@10	0.094	0.244	0.093	0.140	0.147	0.151*	0.229*	0.120*	0.152
TextRank									
F1@5	0.181	0.286	0.185	0.230	0.217	0.022*	0.012*	0.120*	0.157
F1@10	0.151	0.339	0.160	0.216	0.226	0.030*	0.026*	0.181*	0.166
YAKE									
F1@5	0.141*	0.204*	0.215*	0.159*	0.151*	0.105*	0.109*	0.106*	0.149
F1@10	0.146*	0.223*	0.196*	0.196*	0.212*	0.118*	0.135*	0.132*	0.170
RaKUn									
F1@5	0.177*	0.101*	0.127*	0.224*	0.167*	0.168*	0.225*	0.189*	0.172
F1@10	0.160*	0.108*	0.106*	0.193*	0.159*	0.139*	0.185*	0.172*	0.153
Key2Vec									
F1@5	0.080*	0.121*	0.068*	0.109*	0.081*	0.126*	0.158*	0.062*	0.101
F1@10	0.090*	0.181*	0.082*	0.121*	0.126*	0.116*	0.145*	0.078*	0.117
EmbedRank									
F1@5	0.135*	0.345*	0.149*	0.173*	0.189*	0.063*	0.081*	0.219*	0.169
F1@10	0.134*	0.394*	0.158*	0.190*	0.217*	0.057*	0.074*	0.246*	0.184
Supervised algorithms									
KEA									
F1@5	0.046	0.022	0.018	0.073	0.068	/	/	/	/
F1@10	0.044	0.022	0.017	0.071	0.065	/	/	/	/
Maui									
F1@5	0.005	0.035	0.005	0.004	0.011	/	/	/	/
F1@10	0.005	0.046	0.007	0.006	0.014	/	/	/	/
Semi-supervised CopyRNN									
F1@5	0.308	0.326	0.296	0.356	0.322	/	/	/	/
F1@10	0.245	0.334	0.240	0.320	0.294	/	/	/	/
CopyRNN									
F1@5	0.317	0.244	0.305	0.376	0.318	0.406*	0.256*	0.083	0.288
F1@10	0.273	0.289	0.266	0.352	0.318	0.393	0.246	0.105	0.280
CatSeqD									
F1@5	0.348	0.276	**0.325**	**0.374**	**0.327**	0.424*	0.238*	0.063*	0.297
F1@10	0.298	0.333	0.285	**0.366**	**0.352**	0.424*	0.238*	0.063*	0.295
CorrRNN									
F1@5	/	/	0.318	0.361	0.320	/	/	/	/
F1@10	/	/	0.278	0.335	0.320	/	/	/	/
GPT-2									
F1@5	0.275*	0.413*	0.253*	0.318*	0.257*	0.421*	0.331*	0.298*	0.321
F1@10	0.278*	0.469*	0.253*	0.323*	0.278*	0.423*	0.336*	0.312*	0.334
GPT-2 + BiLSTM-CRF									
F1@5	**0.355***	**0.462***	0.287*	0.329*	0.246*	0.478*	**0.386***	**0.333***	0.360
F1@10	**0.360***	0.524*	0.288*	0.336*	0.274*	0.479*	**0.389***	0.371*	0.378
TNT-KID									
F1@5	0.336*	0.460*	0.310*	0.350*	0.283*	**0.485***	0.359*	0.318*	**0.363**
F1@10	0.338*	**0.536***	**0.320***	0.358*	0.337*	**0.485***	0.361*	**0.373***	**0.389**

关键词提取+跨语言关键词检测

近年来，中译英、英译中等“翻译抄袭”作为一种较为隐蔽的学术不端行为，引起了高校、期刊编辑部等越来越多学术科研机构的关注。针对这一现象，知网学术不端文献检测系统在国内外首个实现了跨语言检测功能，为抄袭检测提供了更加严谨科学的检测手段，自发布以来得到论文诚信管理部门的一致认可。

跨语言关键字检测需要模型对于少数据样本训练，有相关论文想法是在多语言语料库上预训练模型，在一种语言上对其进行fine-tune，然后在第二种语言上对模型进行零样本跨语言测试。

[1] Mihalcea R, Tarau P. TextRank: Bringing order into texts[C]. Association for Computational Linguistics, 2004.
[2] https://pdf.hanspub.org/CSA20130100000_81882762.pdf
[3] https://github.com/bitcarmanlee/easy-algorithm-interview-and-practice
[4] 时永宾,余青松.基于共现词卡方值的关键词提取算法[J].计算机工程,2016,42(06):191-195.
[5] 徐涛,蓝传锜.基于卡方统计量的藏文新闻网页关键词提取方法[J].电脑知识与技术,2017,13(26):171-173.
[6] 叶秋永,吴华琼,宋继华. 基于信息增益的中文术语抽取[A]. 中文教学现代化学会.数字化对外汉语教学实践与反思[C].中文教学现代化学会:中文教学现代化学会,2010:7.
[7] 赵京胜,朱巧明,周国栋,张丽.自动关键词抽取研究综述[J].软件学报,2017,28(09):2431-2449.
[8] https://github.com/fxsjy/jieba/blob/master/jieba/analyse/tfidf.py
[9] https://github.com/fxsjy/jieba/blob/master/jieba/analyse/textrank.py
[10] Martinc M , Krlj B , Pollak S . TNT-KID: Transformer-based neural tagger for keyword identification[J]. Natural Language Engineering, 2021:1-40.

北京理工大学
BEIJING INSTITUTE OF TECHNOLOGY

谢谢观看
敬请老师批评指正