广告行业中那些趣事系列31：关键词提取技术攻略以及BERT实践

原创 数据拾光者 数据拾光者 3月27日

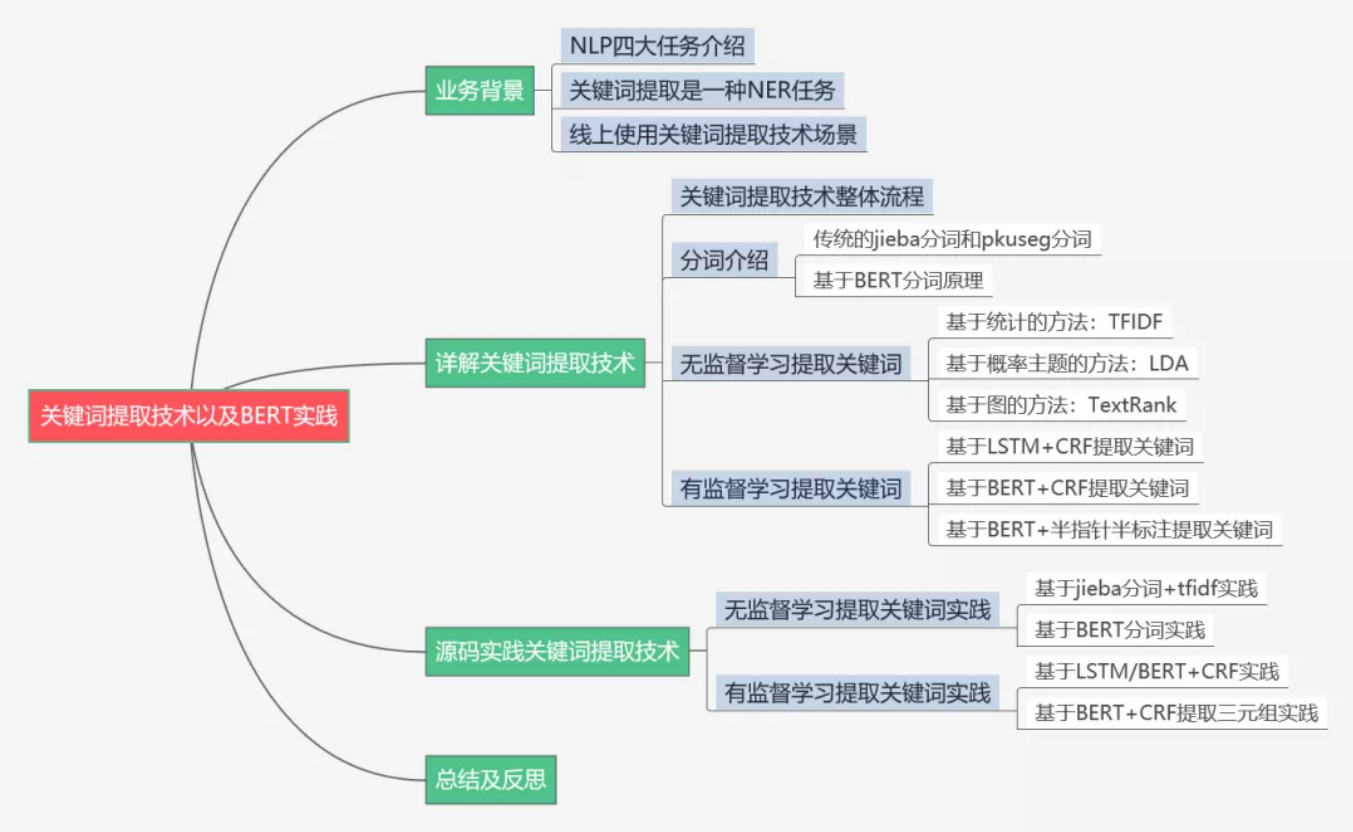
收录于话题
#广告行业中那些趣事系列 41 #NLP 12 #BERT 9 #关键词提取 2

> 导读：本文是“数据拾光者”专栏的第三十一篇文章，这个系列将介绍在广告行业中自然语言处理和推荐系统实践。本篇从理论到实际介绍了NLP领域常见的关键词提取技术，对关键词提取技术感兴趣并希望应用到实际项目中的小伙伴能有所帮助。
>
> 欢迎转载，转载请注明出处以及链接，更多关于自然语言处理、推荐系统优质内容请关注如下频道。
> 知乎专栏：数据拾光者
> 公众号：数据拾光者

封面图：《西游记》里的关键词

摘要：本篇从理论到实际介绍了NLP领域常见的关键词提取技术。首先介绍了业务背景，包括NLP四大任务介绍、关键词提取是一种NER任务、线上使用关键词提取技术场景；然后重点详解了关键词提取技术，包括关键词提取技术整体流程、分词介绍、无监督学习提取关键词、有监督学习提取关键词；最后分别从无监督学习和有监督学习源码实践了关键词提取技术。对关键词提取技术感兴趣并希望应用到实际项目中的小伙伴能有所帮助。

下面主要按照如下思维导图进行学习分享：



01

业务背景

1.1 NLP四大任务介绍

NLP领域有四大任务：**分类、生成、序列标注和句子对标注。**分类任务比较好理解，比如我们要做一个识别用户搜索是否为低俗的分类器，基本上万物皆可分类；生成任务也有很多，比如自动续写小说、诗歌等，之前才分享过文案生成模型等都属于生成任务；句子对标注任务主要会识别两句话是否有关系等等；而序列标注主要就是做命名体识别NER任务，就是从一段文本中抽取想要的内容，关键词提取属于一种NER任务。

1.2 关键词提取是一种NER任务

关键词提取就是从一段文本中抽取具有重要意义的词，在实际业务中应用非常广泛，这里重点在于衡量哪些是关键词，这个和下游任务强相关。下面举一些关键词提取任务的业务示例：第一类**事件主体提取任务**，我们想识别事件主体用于舆情监控领域和金融领域，比如从语句1"*公司A产品出现添加剂，其下属子公司B和公司C遭到了调查*"和语句2"*产品出现问题*"识别出事件主体是"公司A"。这类任务之前有个比赛《CCKS 2019 面向金融领域的事件主体抽取》；第二类**实体识别任务**，比如从语句"*我想去星巴克喝咖啡*"获取商家店铺"*星巴克*"；第三类**事件关系抽取任务**，比如从语句"*九玄珠是在纵横中文网连载的一部小说，作者是龙马*"识别出三元组关系["九玄珠", "连载网站", "纵横中文网"],

["九玄珠", "作者", "龙马"]，第一个三元组的意义是九玄珠的连载网站是纵横中文网，第二个三元组的意义是九玄珠的作者是龙马。如果用符号表示三元组[s, p, o]，相当于要抽取出“s的p是o”这样的关系，这个任务对应的比赛是《2019语言与智能技术竞赛——信息抽取》。

1.3 线上使用关键词提取技术场景

上面是关键词提取任务举例，下面对应到我们实际业务来看下哪些地方需要用到关键词提取技术：首先是**通过关键词圈选人群投放广告。**比如我们需要从用户搜索的query“一刀传奇是谁代言的”中获取关键词“一刀传奇”，然后根据关键词“一刀传奇”来匹配广告。如果完全根据用户搜索来匹配广告那么会存在很多长尾query无法匹配的问题，所以需要提取关键词，通过关键词匹配就可以有效解决长尾query匹配广告的问题了。现在有个传奇游戏相关的广告主购买了我们的词包“一刀传奇”，那么只要用户搜索包含“一刀传奇”关键词那么我们就会匹配对应的广告。本质是根据关键词来圈选人群投放广告；

再比如我们的文案生成模型，会根据广告主选择的行业标签和关键词来生成对应的文案，通常情况下我们希望生成的广告文案是包含关键词的，所以这里需要**提取关键词作为生成条件构建基于seq2seq任务的文案生成模型。**关于文案生成模型相关的介绍可以看下我之前写过的一篇文章《广告行业中那些趣事系列29：基于BERT构建文案生成模型》；

还有关于**搜索召回任务**，搜索场景下根据query召回app广告的query-app任务中会构建DSSM双塔模型，包括query塔和app塔，其中app塔需要获取app对应的关键词作为特征来增加query-app的匹配度。关于DSSM双塔模型小伙伴也可以看下我之前写过的一篇文章《广告行业中那些趣事系列10：推荐系统中不得不说的DSSM双塔模型》。上面这些都是我们实际业务中需要使用关键词提取技术的场景，所以关键词提取这块需要重点学习。

02

详解关键词提取技术

2.1 关键词提取技术整体流程

关键词提取技术整体来看分成两步，第一步是**获取文本的候选词**，第二步则是**对候选词进行打分。**输出的关键词是候选词中得分比较高的。整体流程如下图所示：

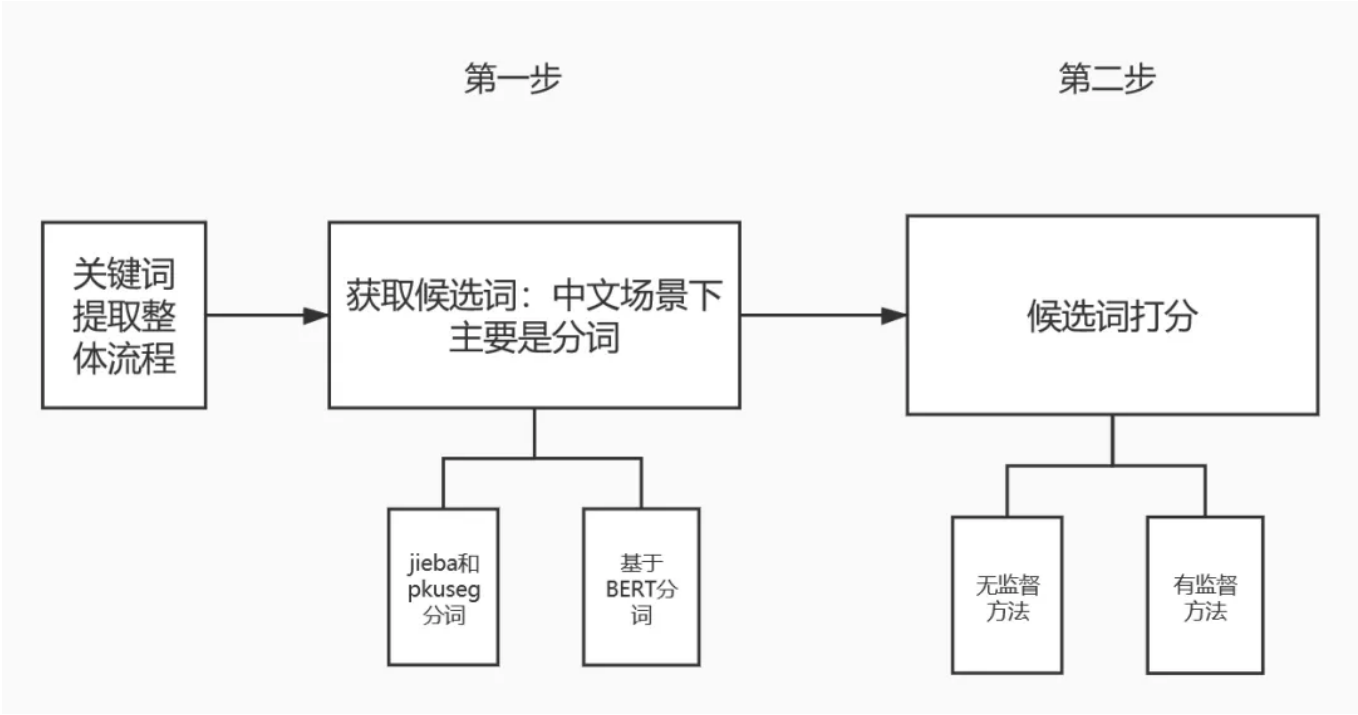


图1 关键词提取整体流程

第一步获取文本的候选词，对于中文场景下最重要的是分词。因为中英两种语言存在非常大的差异，英文本身的最小粒度是词语，通过空格区分；而中文最小粒度是字，所以**获取候选词的前提是需要进行分词**，分词之后进一步获取候选词。

第二步对候选词进行打分，主要分成有监督学习和无监督学习两大类。无监督学习的**优势在于不需要标注数据集，具有一定的普适性**，应用范围较广。但是缺点也很明显，无监督学习的模型效果通常要弱于有监督学习任务，因为有监督学习可以利用标注的数据集获取有用的知识信息，任务也更具有针对性。有监督学习的优点和缺点则和无监督学习刚好相反，不用赘述。

2.2 分词介绍

2.2.1 传统的jieba分词和pkuseg分词

传统分词工具中比较常见的是jieba分词。作为优秀的第三方开源中文分词库，因为简单有效所以被广泛使用。对于大多数NLPer可能用的最多的分词工具就是jieba了，这里不细讲，小伙伴们只需要理解这是一款常用的中文分词工具就行了。之前参加公司比赛的时候主要任务是识别低俗文本，使用传统的文本分类模型比如TextCNN等需要进行分词，尝试了jieba分词和北大开源的pkuseg分词，下面是对比的结果：

模型	分词	precision	recall	fscore	训练时间(min)
TextCNN	jieba	0.824442	0.824442	0.824442	18
TextCNN	pkuseg	0.829025	0.829025	0.829025	18.5
FastText	jieba	0.830915	0.830915	0.830915	14.2
FastText	pkuseg	0.831558	0.831558	0.831558	11.7
TextRCNN	jieba	0.831397	0.831397	0.831397	32.8
TextRCNN	pkuseg	0.833126	0.833126	0.833126	32.6

图2 jieba和pkuseg分词对分类模型效果的影响

在低俗文本分类任务对比结果中可以发现**pkuseg分词效果整体要优于jieba分词**。jieba分词这一类传统分词工具的优势在于简单，普适性广，可以方便的应用到下游各类任务中。

2.2.2 基于BERT分词原理

介绍完传统的分词工具，下面重点说下如何使用BERT进行分词，毕竟我最喜欢的就是万金油的技术。BERT是一种预训练+微调的两阶段模型，因为效果好应用范围广所以被广泛应用到工业界和学术界，其中最重要的原因就是通过预训练学习到海量的语言学知识。那么我们是否可以利用预训练学习到的海量语言学知识来进行中文分词呢？答案是可以的。ACL2020的一篇论文《Perturbed Masking:Parameter-free Probing for Analyzing and Interpreting BERT》提出了一种利用Masked Language Model(MLM)来分析和解释BERT的思路，利用这种思路我们可以用BERT进行分词。

语句是由字组成的序列x=[x1,x2,...xn]，那么我们可以构建nXn的**相关性矩阵T**。通过计算相邻两个字的相关性，然后设置阈值，就可以达到分词的目的。比如"我喜欢吃苹果"这句话我们可以构建6X6的相关性矩阵，每个字相比于其他字都会计算一个相关性值，然后设置一个阈值，当相关性低于某个阈值我们就可以进行切分。**关键是如何衡量相邻两个字之间的相关性**，可以**使用互信息**。对BERT模型来说我们主要通过MLM来衡量相邻两个字之间的相关性。下面通过一张图来说明：

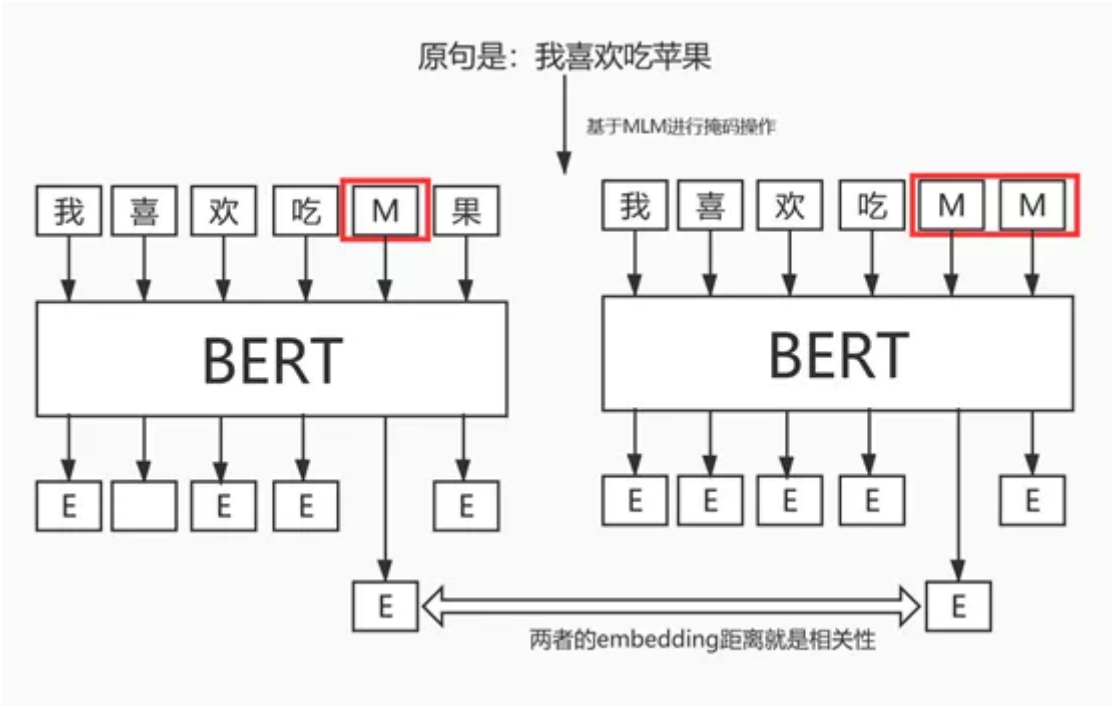


图3 基于BERT的MLM计算相邻两字的相关性

现在有一句话“*我喜欢吃苹果*”，对这句话先将“苹”进行掩码，经过BERT之后会得到字粒度的向量，这里假如“苹”对应的向量是v1；然后对同一句话将“苹”和“果”同时进行掩码，再经过BERT之后也会得到字粒度的向量，假如“苹”对应的向量是v2；最后计算v1和v2的距离，距离越近相关性越好。通过这种方式就可以得到相邻字之间的相关性信息，然后根据相关性信息设置阈值即可进行分词。对应到论文的思路来说，**这两句话的区别在于第一句话只对“苹”进行掩码操作，第二句话对“苹”和“果”同时进行掩码操作，而通常情况下一句话中掩码的字数越多那么模型预测的就越不准，因为可用的信息变少了。所以第一句话得到的“苹”对应的embedding比第二句话“苹”对应的embedding要准，而第二句话相比于第一句话多掩码了“果”字，那么就可以用v1和v2的距离来代表“苹”和“果”的相关性。**最后对比下jieba和BERT分词的效果：

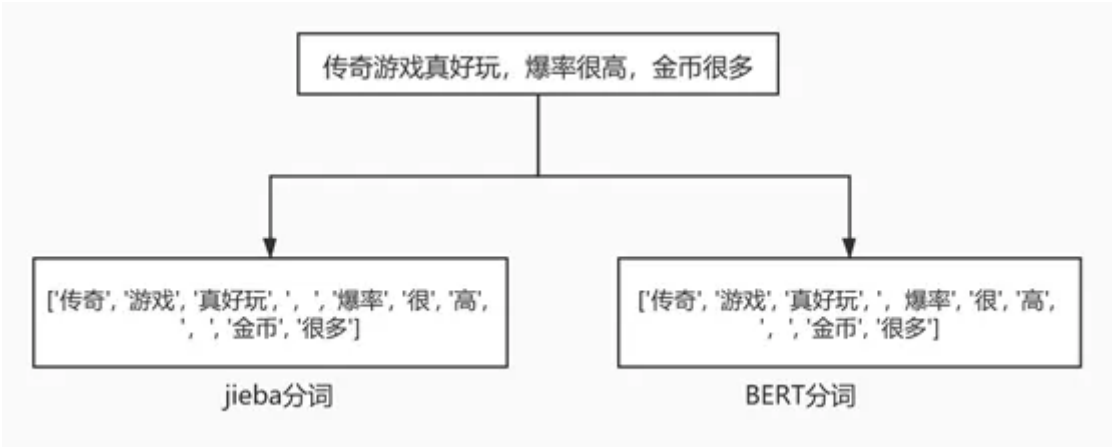


图4 对比jieba分词和BERT分词的效果

通过上图发现jieba分词和BERT分词的区别在于“***爆率***”这里，单从图中看jieba更加合理一点，不过BERT可以设置阈值进行切词，后面效果也是可期的。相比于jieba来说，BERT还有自己独特的优

势：我们**可以用自己业务数据来再训练BERT，使得BERT的切词效果和下游任务有一定的相关性，最终的切词效果也会更好。**

2.3 无监督学习提取关键词

实际工作中无监督学习提取关键词主要分成**基于统计、基于主题概率和基于图的方法。**

2.3.1 基于统计的方法

基于统计的方法主要是利用文档中词语的统计信息来抽取关键词，计算的量化指标主要有基于**词权重**、**词位置**以及**词关联信息。**基于统计的方法优点在于简单、易于实现，不需要标注数据集，泛化性较强。

基于统计的方法主要代表是词频逆文档频率TFIDF算法。TFIDF主要用来**衡量一个词对文档的区分程度**，关于TFIDF算法的原理非常简单，咱们通过一个例子来解释。一般情况下一段文本中出现次数越多的词越可能是关键词，但是对于一些常见的比如“你”、“我”、“他”之类的词可能在很多文档中都多次出现，但是这些词却不属于关键词。所以**我们的目标是要找到那些在当前文档中出现次数很多，但是在大多数文档中出现次数很少的词作为当前文档的关键词。**对应到TFIDF算法就包括两部分，第一部分是计算词频TF，这部分就是计算各个词在当前文档中出现的次数；第二部分是计算逆文档频率IDF，这部分是计算词在文档库中的普遍程度，作用是如果一个词在大多数文档中都出现，那么对应的IDF的值就会比较小，说明这个词**大概率是通用性比较强但区分性比较差的混子词。**评价一个词是当前文档中的关键词是需要在当前文档中出现的次数比较大(TF比较大)，同时在大多数文档中出现次数比较少的词(IDF比较大)。对应的数学公式就是如下所示：

TFIDF计算公式：

$$TFIDF = \underbrace{TF}_{\text{词频}} \times \underbrace{IDF}_{\text{逆文档频率}} = \frac{n_{ij}}{\sum_{k} n_{kj}} \times \log\left(\frac{|D|}{1+|D_i|}\right)$$

图5 TFIDF计算公式

关于TFIDF计算公式内部细节这里不再赘述，感兴趣的小伙伴可以自行查阅。因为TFIDF算法完全是基于数学统计的，所以不需要标注数据集，同时本身非常简单，通用性很好，可以作为简单的baseline。尤其对于现在很多复杂的业务场景很多简单的方法往往能达到很不错的线上效果。TFIDF算法的缺点主要有以下三个方面：第一，单纯以词频衡量一个词的重要性不够全面；第二，无法体现词的位置、词性和关联信息等特尔正；第三，无法反应词汇的语义信息。

2.3.2 基于概率主题的方法

上面说到TFIDF这种基于统计的方法缺点是无法反应词语的语义信息，针对这个问题主要有基于概率主题的方法。**基于概率主题的方法是语义挖掘的核心**，主题模型认为文档是有很多主题组成的，**文档**

既是主题的分布也是关键词的分布。常见的基于概率主题的方法主要由LSA、LDA(潜在狄利克雷分布)算法等。下面是主题模型映射示意图：

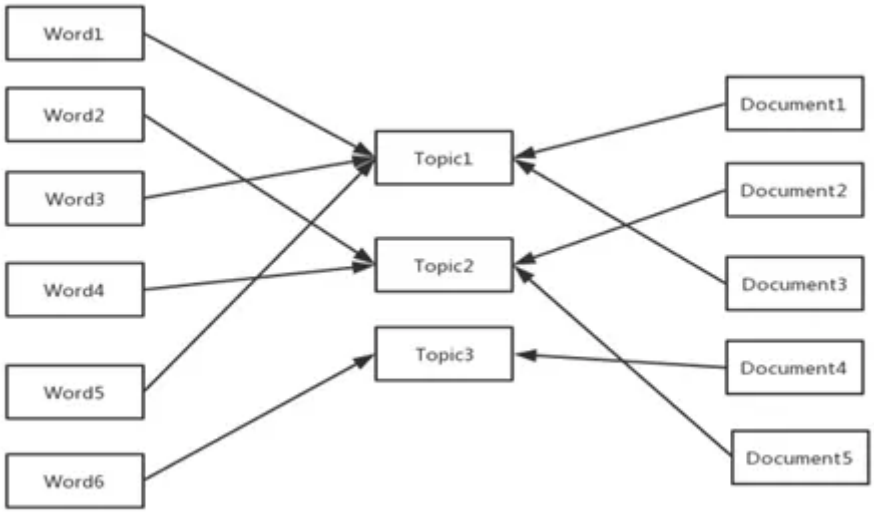


图6 主题模型映射示意图

下面以LDA模型为例讲解基于概率主题的方法，下面是LDA的直观现象图：

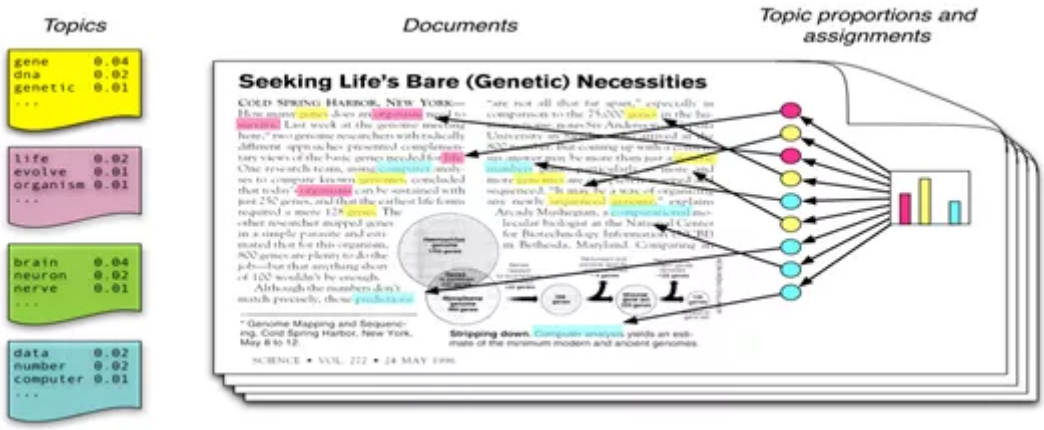
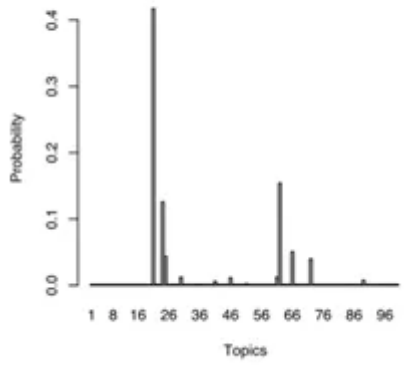


图7 LDA直观现象

可以发现上图对应的文档中会分成很多主题，这些主题分别是黄色、紫色、绿色和蓝色等，而每个主题对应各自的关键词，下面是各主题和关键词对应的关系图：



"Genetics"	"Evolution"	"Disease"	"Computers"
human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers
sequence	origin	bacterial	system
gene	biology	new	network
molecular	groups	strains	systems
sequencing	phylogenetic	control	model
map	living	infectious	parallel
information	diversity	malaria	methods
genetics	group	parasite	networks
mapping	new	parasites	software
project	two	united	new
sequences	common	tuberculosis	simulations

图8 各主题和关键词对应的关系图

上图中左边部分是文档的主题分布概率，其中头部主要包括“genetics”、“evolution”、“disease”和“computers”四个主题，每个主题会对应一定的关键词。通过这种基于主题概率分布的方法，LDA模型有以下优点：首先，可以获得文本语义相似的关系，可以一定程度上解决多义词的问题；然后，LDA还可以去除文档中噪音的影响；其次，LDA是一种无监督的方法，可以完全自动化，不需要人工标注数据集，可以直接通过模型得到概率分布；最后，LDA和语言无关，模型的应用范围更广。

2.3.3 基于图的方法

基于图的方法理论基础在于**人类语言是复杂网络，具有小世界特性和无标度特性，关键词提取就是寻找语言网络中起中心作用的词**，其中有代表性的算法是TextRank。TextRank算法的基本思想来源于谷歌的PageRank算法。PageRank算法是一种网页排名算法，基本的思想：**网页的重要性得分主要由链接质量和链接数量决定。**通过下图说明PageRank算法：

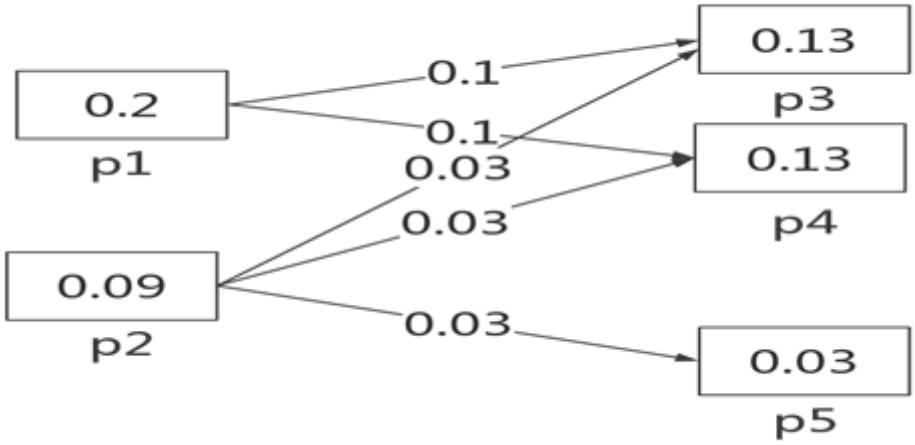


图9 PageRank算法说明图

上图中总共有五个网页p1-p5。假如p1的得分为0.2，因为p1有两个下游链接网页p3和p4，那么会将0.2平均分到p3和p4；P2的得分为0.09，对应三个下游链接网页p3、p4、p5，那么综合计算下来p3和p4的得分就是0.1+0.03=0.13，p5的得分仅为0.03。网页链接的得分最终取决于链接的数量和质量，上游的网页链接数量越多，质量越高(这里指得分)那么该网页链接就是相对中心的网络点，重要性越高。

TextRank算法的思想虽然来源于PageRank，但也有不同之处：**PageRank是有向无权图**，而**TextRank是有权图**，这里**权重是两个句子间的相似性。**下面是TextRank计算公式：

$$S(V_i) = (1-d) + d \times \sum_{j \in In(V_i)} \left(\frac{w_{ji}}{Out(V_j)} \times S(V_j) \right)$$

权重

加入阻尼指数

图10 TextRank计算公式

总结下，基于图方法的TextRank算法具有以下特点：

- 无需训练数据，节省了大量成本
- 适应性强。无监督学习方法，具有很强的适应能力和扩展能力，对文本没有主题方面的限制
- 速度快，虽然是矩阵运算，但是收敛速度快
- 一定程度上考虑了文本结构，实际效果比TFIDF好
- TextRank更擅长处理长文本，因为短文本词汇信息弱，构建图不理想
- TextRank仍然更倾向于较为频繁的词作为关键词

2.4 有监督学习提取关键词

上面介绍了无监督学习提取关键词，下面介绍有监督学习提取关键词。前面也说过关键词提取属于NER任务，NER任务虽然属于一个历史悠久的NLP任务，但是自从**2015年LSTM+CRF出世因为模型本身和任务匹配度非常高基本成为主流**。后来BERT模型出来之后，模型结构就变成了**BERT(+LSTM)+CRF结构**。

2.4.1 基于LSTM+CRF提取关键词

LSTM+CRF模型是2015年在论文《BidirectionalLSTM-CRF Models for Sequence Tagging》中被提出来的，模型结构如下图所示：

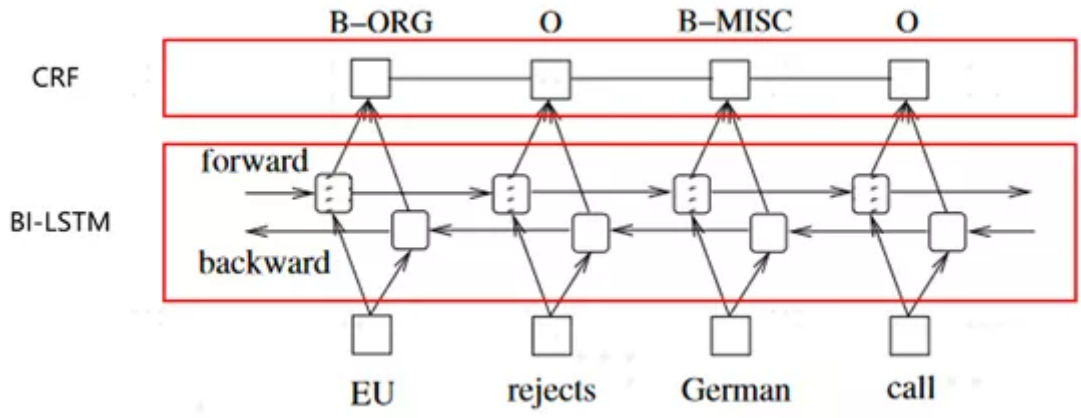


图11 LSTM+CRF模型结构图

模型整体分成两部分，**第一部分是双向LSTM**，包括两个LSTM cell，其中一个负责从左到右得到第一层表征向量L，另一个负责从右到左得到第二层表征向量R，然后将两层向量相加得到LSTM部分最终的向量V；**第二部分是CRF**，将向量V经过CRF层会得到最终的结果，对于序列标注任务来说基本

上每个词都会有对应的输出，**常用的表示序列标注结果的方法有BIO标记法和BIOES标记法两种**。这里不再细讲。可以这么说在BERT出来之前序列标注任务主要是使用LSTM+CRF这种模型结构。

2.4.2 基于BERT+CRF提取关键词

后来BERT横空出世，因为BERT超强的编码能力所以后面主要用BERT+CRF来解决序列标注任务，模型结构主要是将LSTM换成了BERT，下面是模型结构图：

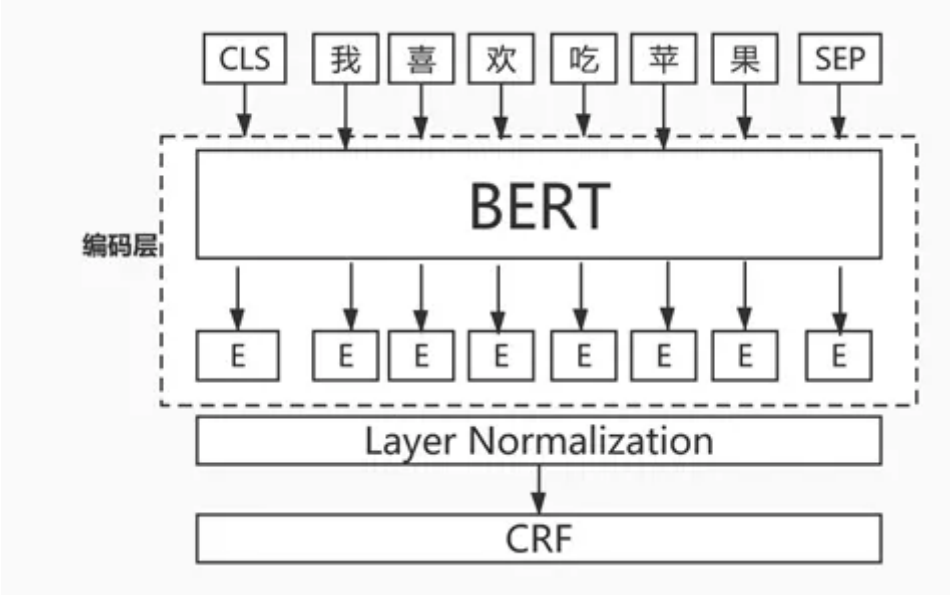


图12 BERT+CRF模型结构图

使用BERT+CRF模型时**需要注意BERT部分和CRF部分需要使用不同的学习率，BERT部分使用较小的学习率，CRF部分使用较大的学习率。**原因在于BERT进行预训练之后，模型的拟合能力很强，针对下游任务进行微调时只需要设置很小的学习率就可以充分拟合训练数据，太大反而可能不收敛。如果CRF部分使用和BERT一样的学习率可能导致CRF层训练不充分，所以CRF部分需要设置较大的学习率才能学习充分，这个也是经过实验证明的。

2.4.3 基于BERT+半指针半标注提取关键词

除了基于BERT+CRF提取关键词，苏神还分享了一种基于BERT+半指针半标注模型用于提取关键词，因为之前写过一篇文章《广告行业中那些趣事系列17：实战基于BERT和指针网络的实体抽取》，里面有详细的源码讲解，整体效果还不错，这里不再赘述，下面是模型结构图：

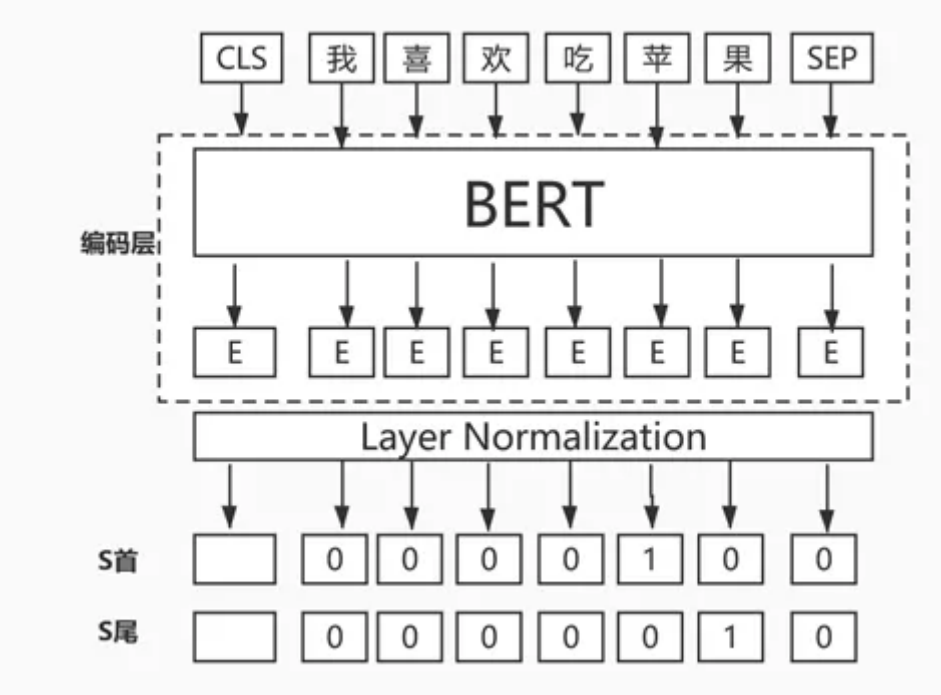


图13 基于BERT+半指针变标注模型结构

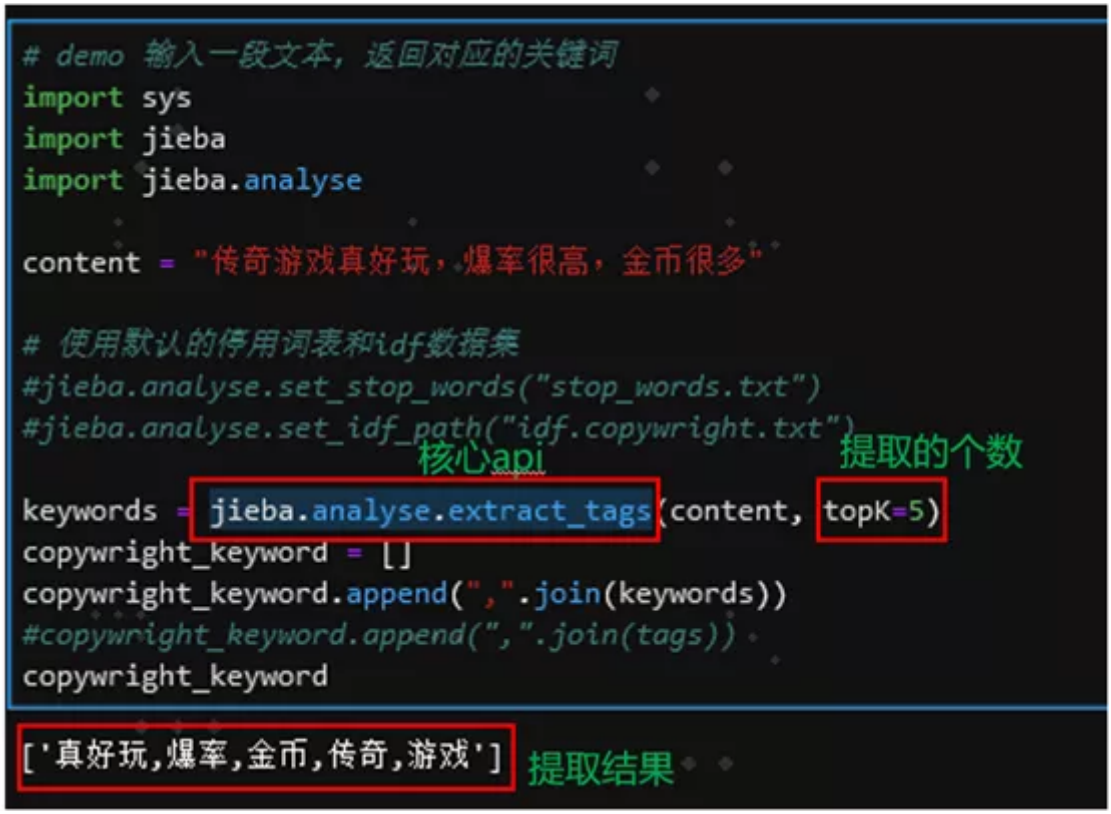
03

源码实践关键词提取技术

3.1 无监督学习提取关键词实践

3.1.1 基于jieba分词+tfidf实践

基于jieba分词+tfidf源码实践关键词抽取非常简单，主要是用jieba.analyse提供的extract_tags接口就行了，下面是源码及关键词提取效果：




```
# demo 输入一段文本，返回对应的关键词
import sys
import jieba
import jieba.analyse

content = "传奇游戏真好玩，爆率很高，金币很多"

# 使用默认的停用词表和idf数据集
#jieba.analyse.set_stop_words("stop_words.txt")
#jieba.analyse.set_idf_path("idf.copywright.txt")

keywords = jieba.analyse.extract_tags(content, topK=5)
copywright_keyword = []
copywright_keyword.append(",".join(keywords))
#copywright_keyword.append(",".join(tags))
copywright_keyword

['真好玩,爆率,金币,传奇,游戏']
```

图14 基于jieba分词+tfidf源码及提取效果

上图中需要注意的是**设置提取个数topK=5**，那么最终的提取结果就只包含五个关键词，整体来看提取结果还是不错的。这里需要注意的是**可以根据自身业务需要调整停用词表和idf逆文档频率文件中关键词的idf值。**下面是核心接口extract_tags函数的源码：

```python
def extract_tags(self, sentence, topK=20, withWeight=False, allowPOS=(), withFlag=False):
    # (1) 中文分词
    if allowPOS:
        allowPOS = frozenset(allowPOS)
        words = self.postokenizer.cut(sentence)
    else:
        words = self.tokenizer.cut(sentence)

    # (2) 计算词频TF
    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
        if len(wc.strip()) < 2 or wc.lower() in self.stop_words:
            continue
        freq[w] = freq.get(w, 0.0) + 1.0
    total = sum(freq.values())

    # (3) 计算IDF
    for k in freq:
        kw = k.word if allowPOS and withFlag else k
        freq[k] *= self.idf_freq.get(kw, self.median_idf) / total

    # (4) 排序得到关键词集合
    if withWeight:
        tags = sorted(freq.items(), key=itemgetter(1), reverse=True)
    else:
        tags = sorted(freq, key=freq.__getitem__, reverse=True)
    if topK:
        return tags[:topK]
    else:
        return tags
```

图15 extract_tags函数的源码

extract_tags函数主要完成了四个工作，分别是中文分词、计算词频TF、计算IDF和最终得到TFIDF算法排序之后的关键词集合。

3.1.2 基于BERT分词实践

BERT分词源码实践主要是基于苏剑林开源的bert4keras，原理上面已经讲解，github开源地址如下：

https://github.com/bojone/perturbed_masking/blob/master/word_segment.py

3.2 有监督学习提取关键词实践

3.2.1 基于LSTM+CRF和BERT+CRF的实践

有监督学习提取关键词实践分享一个非常不错的开源项目，作者对LSTM+CRF、BERT+CRF等都做了完整的实验，推荐小伙伴们可以关注学习下。下面是开源项目地址：

https://github.com/wavewangyue/ner/tree/master

3.2.2 基于BERT+CRF提取三元组实践

分享一个苏神开源的用bert4keras提取三元组的开源项目地址：

https://github.com/bojone/bert4keras/blob/master/examples/task_relation_extraction.py

04

总结及反思

本篇从理论到实际介绍了NLP领域常见的关键词提取技术。首先介绍了业务背景，包括NLP四大任务介绍、关键词提取是一种NER任务、线上使用关键词提取技术场景；然后重点详解了关键词提取技术，包括关键词提取技术整体流程、分词介绍、无监督学习提取关键词、有监督学习提取关键词；最后分别从无监督学习和有监督学习源码实践了关键词提取技术。对关键词提取技术感兴趣并希望应用到实际项目中的小伙伴能有所帮助。

05

参考资料

[1] Bidirectional LSTM-CRF Models for Sequence Tagging
[2] Chinese NER Using Lattice LSTM
[3] Perturbed Masking: Parameter-free Probing for Analyzing andInterpreting BERT
[4] 苏剑林. (Jun. 10, 2020). 《无监督分词和句法分析！原来BERT还可以这样用》[Blog post]. Retrieved fromhttps://www.kexue.fm/archives/7476

最新最全的文章请关注我的微信公众号或者知乎专栏：数据拾光者。

码字不易，欢迎小伙伴们点赞和分享。

喜欢此内容的人还喜欢

广告行业中那些趣事系列35：NLP场景中的对比学习模型SimCSE