1.TF-IDF和关键词提取

作为提取关键词的最基本、最简单易懂的方法，首先介绍下TF-IDF。

判断一个词在一篇文章中是否重要，一个容易想到的衡量指标就是词频，重要的词往往会在文章中多次出现。但另一方面，不是出现次数多的词就一定重要，因为有些词在各种文章中都频繁出现，那它的重要性肯定不如那些只在某篇文章中频繁出现的词重要性强。从统计学的角度，就是给予那些不常见的词以较大的权重，而减少常见词的权重。IDF（逆文档频率）就是这个权重，TF则指的是词频。

> TF=（词语在文章中出现次数）/（文章总词数）
>
> IDF=log (语料库文档总数/(包含该词的文档数+1))
>
> TF - IDF = TF * IDF

摘取一个博客中的一个例子[1]

	包含该词的文档数（亿）	IDF	TF-IDF
中国	62.3	0.603	0.0121
蜜蜂	0.484	2.713	0.0543
养殖	0.973	2.410	0.0482

“中国”在文章中的频率并不比“蜜蜂“和”养殖“低，但因其在各种文章中都会频繁出现，因此其逆文档频率较低，不会被识别成本文的关键词。

TF-IDF虽然非常简单，但却很经典有效，而且速度很快，有的场景中会提升第一段和最后一段的文本权重，因为文章的关键词往往会在开头和结尾段频繁出现。但TF-IDF只是仅从词频角度挖掘信息，并不能体现文本的深层语义信息。

2.topic-model和关键词提取

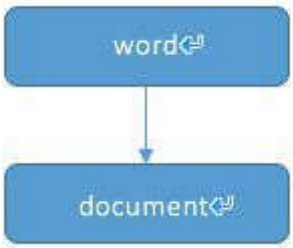
如果说TF-IDF只能从词频角度挖掘信息，那么如何挖掘更深层次的信息呢？这就是topic-model想要完成的任务。

举个例子，以下四个句子：

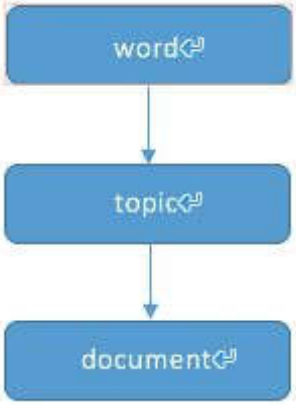
> 1.I
> ate a banana and spinach smoothie for breakfast
>
> 2.I
> like to eat broccoli and bananas.
>
> 3.Chinchillas
> and kittens are cute.
>
> 4.My
> sister adopted a kitten yesterday.

仅从词语角度分析，1.2句banana是重复出现的，3.4句kitten是重复出现的。但其实可以发现1.2句主要跟食物有关，3.4句主要跟动物有关，而food、animal两个词在四句话里均未出现，有没有可能判断出四句话中所包含的两个主题呢？或者当两篇文章共有的高频词很少，如一篇讲banana，一篇讲orange，是否可以判断两篇文章都包含food这个主题呢？如何生成主题、如何分析文章的主题，这就是topic-model所研究的内容。对文本进行LSA（隐形语义分析）。

在直接对词频进行分析的研究中，可以认为通过词语来描述文章，即一层的传递关系。



而topic-model则认为文章是由主题组成，文章中的词，是以一定概率从主题中选取的。不同的主题下，词语出现的概率分布是不同的。比如”鱼雷“一词，在”军事“主题下出现的概率远大于在”食品”主题下出现的概率。即topic-model认为文档和词语之间还有一层关系。



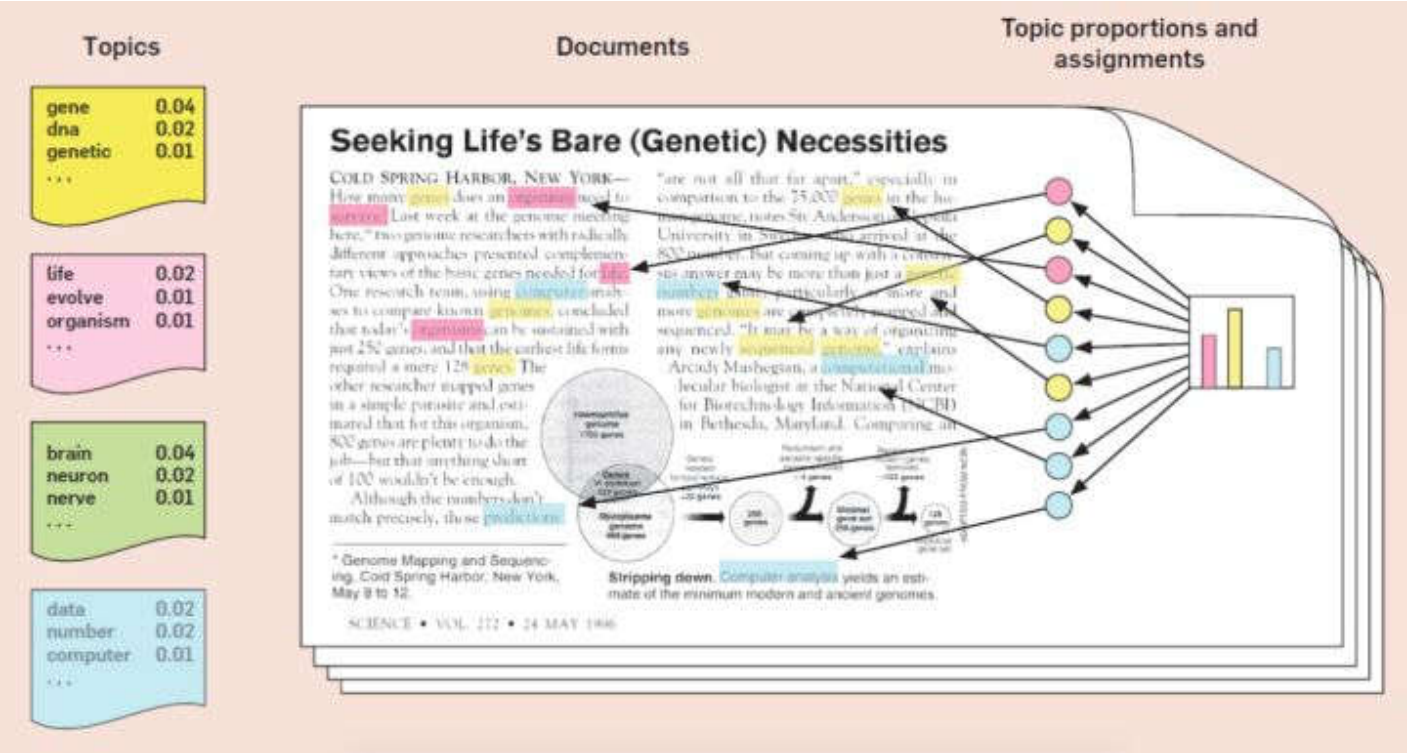
首先假设每篇文章只有一个主题z，则对于文章中的词w，是根据在z主题下的概率分布p(w|z)生成的。则在已经选定主题的前提下，整篇文档产生的概率是 $\prod_{w} p(w \vee z)$

而这种对每篇文章只有一个主题的假设显然是不合理的，事实上每篇文章可能有多个主题，即主题的选择也是服从某概率分布p(t)的因此根据LDA模型，所有变量的联合分布为

$$p(\vec{w}_m, \vec{z}_m, \vec{\vartheta}_m, \underline{\Phi}|\vec{\alpha}, \vec{\beta}) = \overbrace{\underbrace{\prod_{n=1}^{N_m} p(w_{m,n}|\vec{\varphi}_{z_{m,n}}) p(z_{m,n}|\vec{\vartheta}_m) \cdot p(\vec{\vartheta}_m|\vec{\alpha})}_{\text{word plate}} \cdot \underbrace{p(\underline{\Phi}|\vec{\beta})}_{\text{topic plate}}}^{\text{document plate (1 document)}}.$$

φ 表示topic下词的分布，θ 表示文档下topic的分布。N_m 是第m个文档的单词总数。α 和 β 表示词语和topic的概率分布先验参数。而学习LDA的过程，就是通过观察到的文档集合，学习 φ, θ, z 的过程。学习过程参见论文[2]。

下图为一个LDA学习结果的例子



取自[3]

可以看出，topic-model的目的就是从文本中发现隐含的语义维度，在词语和文档之间加入更概括的信息。

3.textrank关键词提取

textrank的则从图模型的角度找文章的关键词，好处在于不用事先基于大量数据进行训练。

其基本思想来自于pagerank算法，pagerank的两条基本思想是，如果一个网页被很多其他网页链接到，说明这个网页比较重要；如果一个网页被一个权值很高的网页链接到，则其重要性也会相应增加。

$$S(V_i) = (1-d) + d * \sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} S(V_j)$$

判断两个网页之间是否有边相连，根据网页中出现的链接，而在textrank中判断两个词间是否存在相关关系，则根据词语的共现关系。实际处理时，取一定长度的窗，在窗内的共现关系则视为有效。

修改的textrank算法

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

4.rake关键词提取

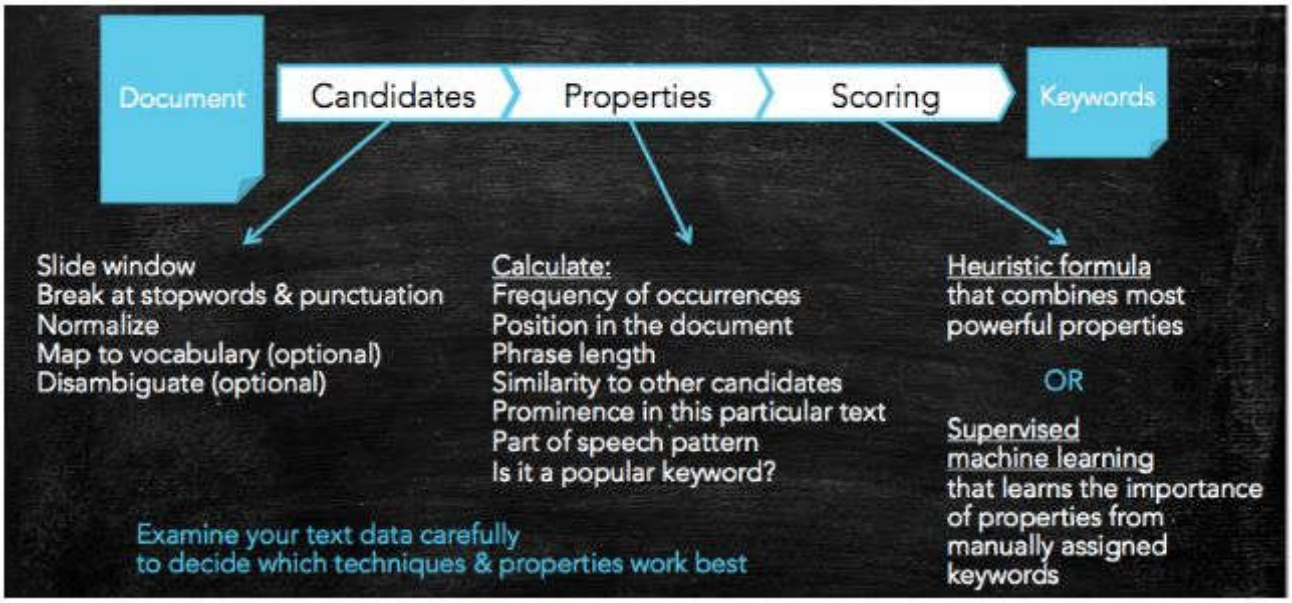
rake算法提取的并不是单一的单词，而是由单词组成的短语。短语的分割由标点符号

Extracted by RAKE
minimal generating sets
linear diophantine equations
minimal supporting set
minimal set
linear constraints
natural numbers
strict inequations
nonstrict inequations
upper bounds

每个短语的得分由组成短语的词累加得到，而词的得分与词的度与词频有关

$$Score = \frac{degree}{frequency}$$

当与一个词共现的词语越多，该词的度就越大。



算法本身很简单也很好理解，也有可直接供使用的python代码：

GitHub - aneesha/RAKE: A python implementation of the Rapid Automatic Keyword Extraction

参考文献

[1]
TF-IDF与余弦相似性的应用（一）：自动提取关键词

[2]
Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. Journal of machine Learning research, 2003, 3(Jan): 993-1022.

[3]
Blei D M. Probabilistic topic models[J]. Communications of the ACM, 2012, 55(4): 77-84.

[4]
Rose S, Engel D, Cramer N, et al. Automatic keyword extraction from individual documents[J]. Text Mining, 2010: 1-20.