

介绍：（to be cont.）

我们采用的词向量模型是Word2Vec模型。绝大部分项目会选择Continuous Bag-of-Words（CBOW）. 但由于我们训练用的语料库较小且关系更为复杂，所以我们选择skip-gram。因为在skip-gram中，每个单词都有一个独立的向量表示，因此每个向量可以专门捕捉该单词在语料库中的不同语义，包括它在不同上下文中的用法。

数据来源：

我们从UN Economic and Social Council (ECOSOC)发布的文章中选取了最近发布的57篇文章使用Natural Language Processing(NLP)技术进行分析。根据关键词来寻找17个SDG之间的network of the relationship.

详细步骤：

1. 目标定位

我们根据联合国的17个SDG，将每个SDG细分为若干个与其相关的关键词，在我们选择57篇文章当中筛选相关关键词的出现频率。

放一个关键词的表格

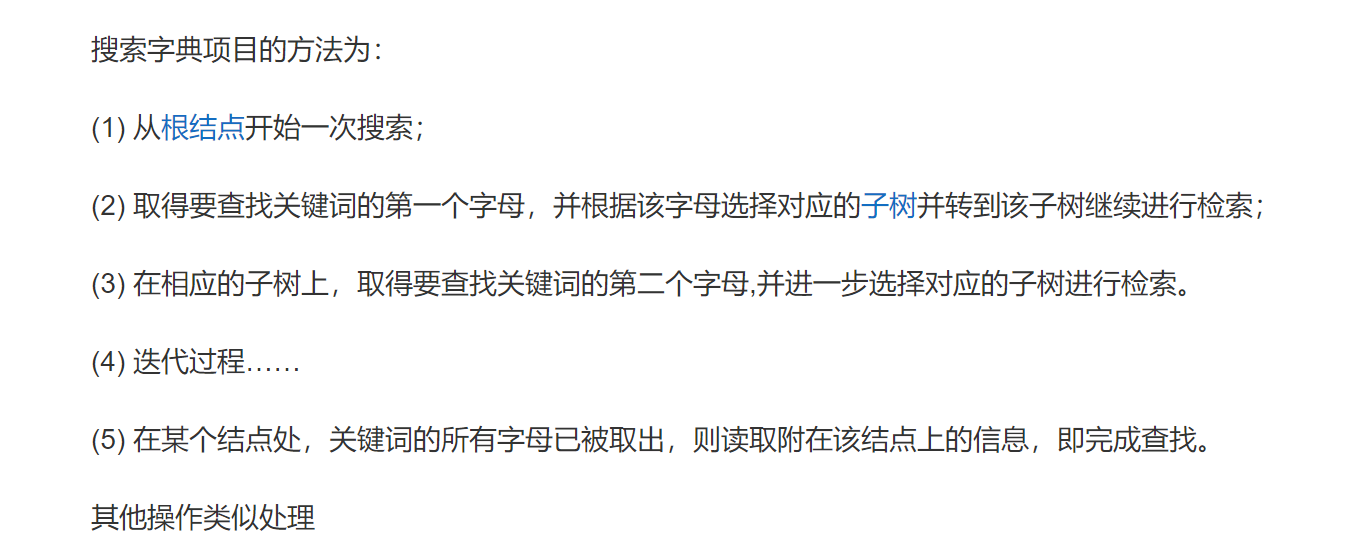
1. 文章预处理

将文章转换为机器可读的格式。对此，我们使用Python中的Natural Language Toolkit (NLTK)库进行预处理，包括去除标点符号，分词，去除停用词等。

我们认为分词算法根据其核心思想主要分为两种，第一种是基于字典的分词，先把句子按照字典切分成词，再寻找词的最佳组合方式；第二种是基于字的分词，即由字构词，先把句子分成一个个字，再将字组合成词，寻找最优的切分策略，同时也可以转化成序列标注问题。归根结底，上述两种方法都可以归结为在图或者概率图上寻找最短路径的问题。

将句子分割成单词或子词，以便计算机可以理解和处理。在英文中，通常是以空格或标点符号为分隔符。而在中文中则需要进行中文分词，将一段连续的中文文本切分成有意义的词语。

这里，我们采用jieba词库进行分词，jieba采用了基于Trie树的分词算法，具有较高的分词速度和较小的内存占用。Trie树是基于以下搜索字典项目(改写成为伪代码)：



综上考虑，我们决定使用jieba 分词。

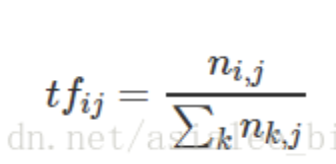
接着，我们还会采用停用词过滤的方法，停用词是指在文本中出现频率非常高，但并没有实际意义的词语，如“的”、“是”、“在”等。过滤掉这些停用词可以降低数据的噪声和冗余，提高后续处理的效率和质量。摆脱停用词对于任何自然语言处理任务都非常有意义。NLTK还允许用户根据自己的需要定制停止词列表。例如，他们可以从列表中添加或删除单词，或者根据自己的文本语料库创建一个新的列表。因此，我们会选择NLTK作为停用词表。

词干提取和词形还原：将单词的不同形态（如时态、复数形式）转换成其原始形式，以便提高文本的准确性和一致性。词干提取是指将单词缩减到其基本形式，如将“running”和“runs”都转换为“run”，而词形还原则是指将单词恢复到其原始形式，如将“am”, “is”, “are”都还原为“be”。

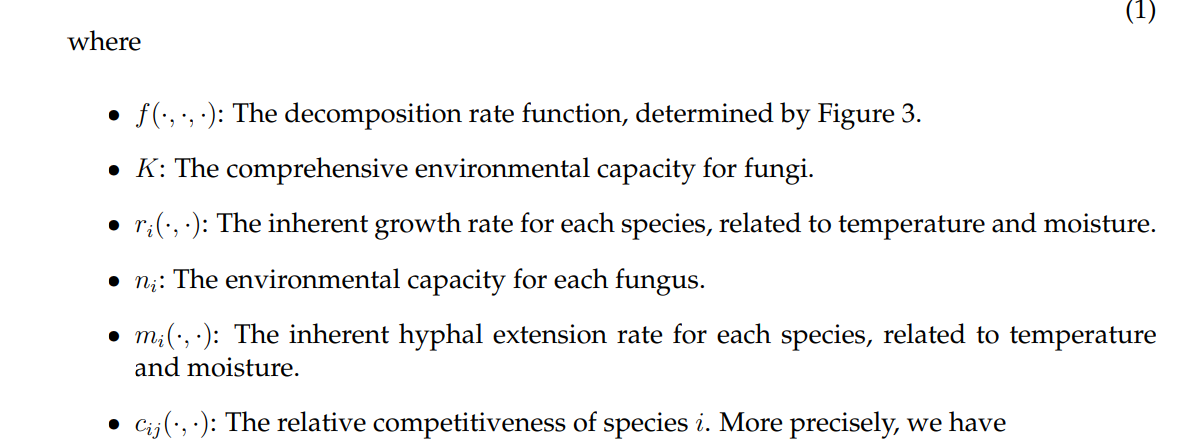
1. 筛选关键词

为了更好的寻找到文章的关键词，需要的文章进行处理。我们采用TF-IDF算法，便于区分字词的重要性, 以此来筛选具有良好的类别区分能力词。

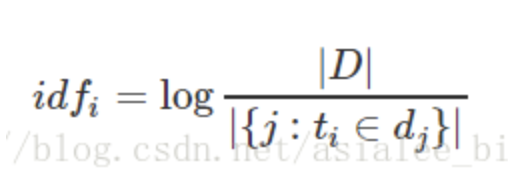
第一步，需要统计词汇出现的数量，也就是计算Term Frequency。在统计到文章词汇数量后，由于文章长短不齐，为了防止文章长短对数据造成一定影响，需要对该数据进行normalized processing。公式如下：



**N（i,j）**是该词在文章中出现的次数，分母则是文章中所有词汇出现的次数总和

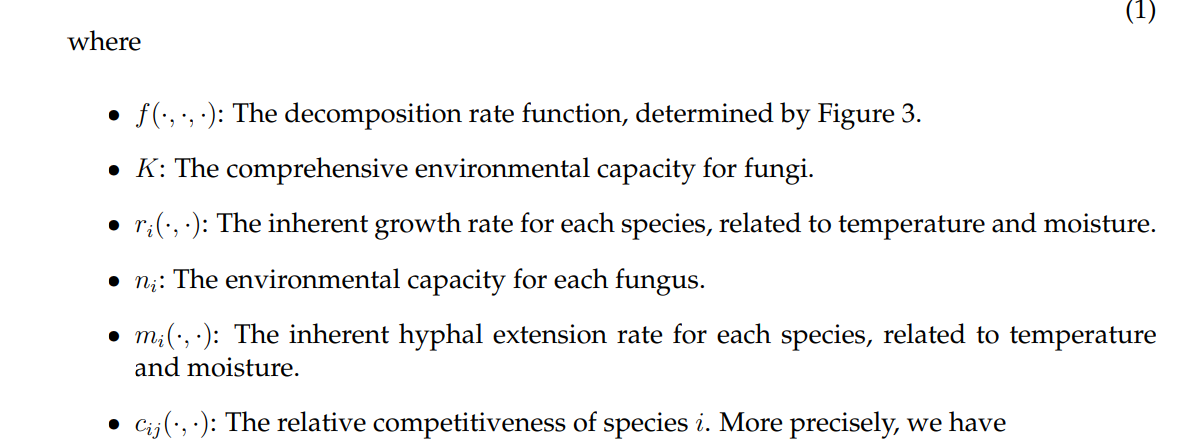


第二步，我们需要计算Inverse Document Frequency（IDF），由**总文件数目除以包含该词语的文件的数目**，**再将得到的商取对数得到**。词汇的IDF越大，说明其类别区分能力越好。IDF计算公式如下：



**ps:(这里分母应该是1+|{j:ti∈dj}|**

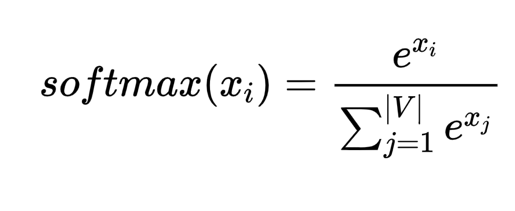
**|D|** **是文章总数**。**|{j:ti∈dj}| 表示包含该词汇的文章数目**



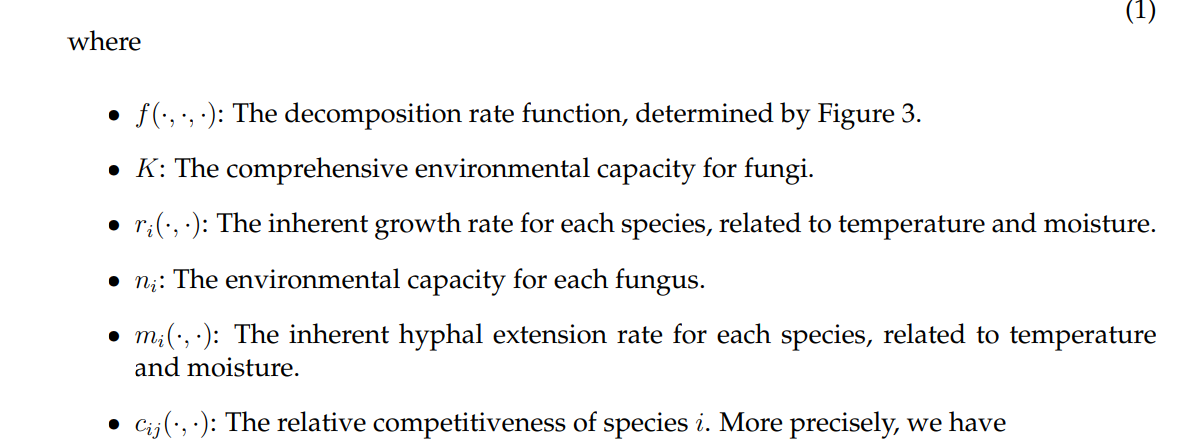
将TF与IDF相乘即可过滤掉常见的词汇。由此，我们可以提取每篇文章的关键词。

1. 训练Word2Vec模型

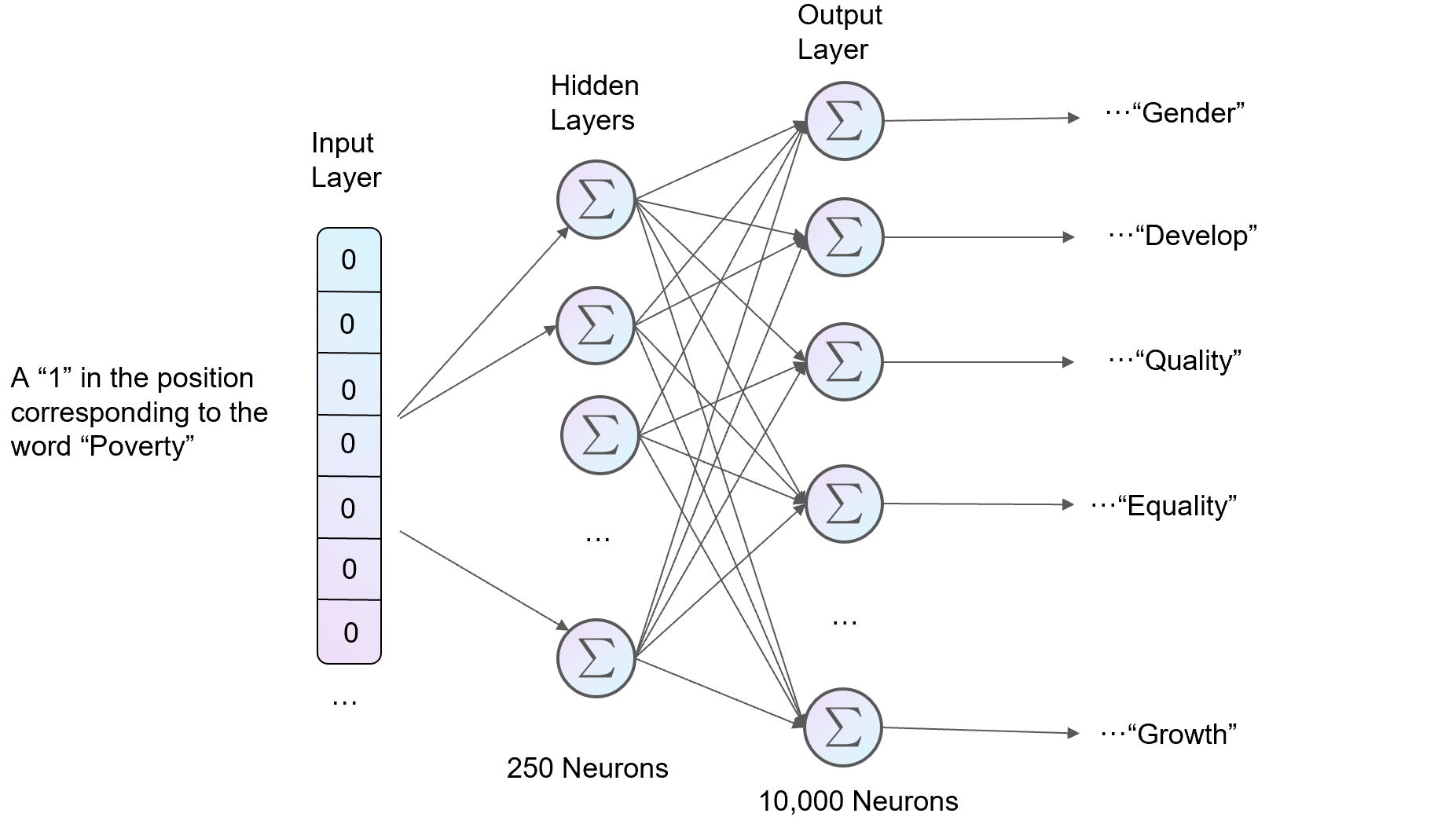
首先使用NLTK库构建词汇表，将每个单词映射到唯一的整数标识。接下来就是训练Word2Vec模型，设置skip-window=4，Hidden Layer=90。为了评估可持续发展目标联合国报告中的重叠与科学文献中可持续发展目标之间的整合之间的一致性，我们收集了Google Scholars及ECOSOC发布的其他文章，以此作为训练数据。我们使用TensorFlow作为工具来训练Word2Vec skip-gram模型。这里需要将上述所有文章作为输入。对外，我们采用one-hot进行编码将所有的字符转化成二进制的编码形式。再将one-hot编码后的词向量，通过神经网络的hidden layer，映射到一个低纬度的空间。再将hidden layer和输出层进行全连接。然后，我们需要将向量转换为概率分布的函数softmax用于计算当前单词与其他单词的相似度，以确定最可能的上下文单词。softmax函数可以表示为：



$x\_i$表示当前单词向量与单词i向量的点积结果，$|V|$表示词汇表中单词的数量



Finally，we got:



模型实现：

