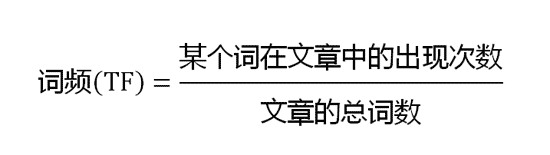
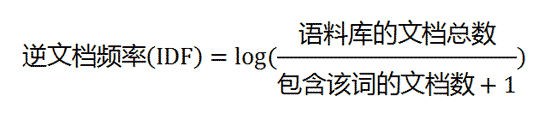
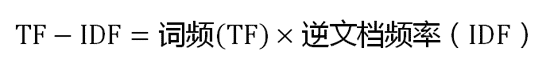
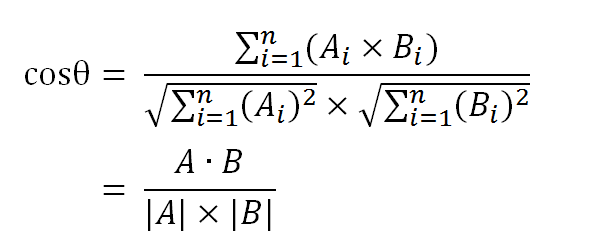
# 余弦相似度算法说明文档

本算法目的是计算两个文本之间的余弦相似度。为达成此目标，我们首先需要计算单词在文本库中各文档的Term Frequency，以及单词在文本库中的Inverse Document Frequency。其次计算文本中单词对应的TF-IDF，形成向量集合后再通过公式计算得到文本之间的余弦相似度。

涉及概念如下：



## 定义的结构与关系如下：

## 相关函数描述：

* ReadFile

参数：path:string

返回：file\_read:string

接收：无

功能：将path代表的文件内容转为string

* CutLine

参数：file\_stream:string

返回：[]string

接收：无

功能：对string类型的文件内容进行分词

* split

参数：singleChar:rune

返回：bool

接收：无

功能：在Cutline中确认分隔符号，可以自行设置

* Load

参数：filePaths:[]string

返回：wordsMap:map[string][]string

接收：无

功能：对于所给文件路径表，遍历加载单词字典map

* GetFeature

参数：wordStream: map[string][]string

返回：TDMap:map[string]map[string]float64, IDFMap:map[string]float64

接收：无

功能：分别计算单词在文本库中各文档的TF，以及单词在文本库中的IDF

* CalculateTFIDFMap

参数：TFMap:map[string]map[string]float64, IDFMap:map[string]float64

返回：TFIDFMap:map[string]map[string]float64

接收：无

功能：将文本中对应单词的TF与单词自身IDF相乘得到TF-IDF

* CalculateVectorCos(fileAPath string, fileBPath string, TFIDFMap map[string]map[string]float64) float64

参数：fileAPath:string, fileBPath:string, TFIDFMap:map[string]map[string]float64

返回：float64

接收：无

功能：由文本中的单词对应的TF-IDF值形成的向量根据公式来计算余弦相似度

* main

参数：无

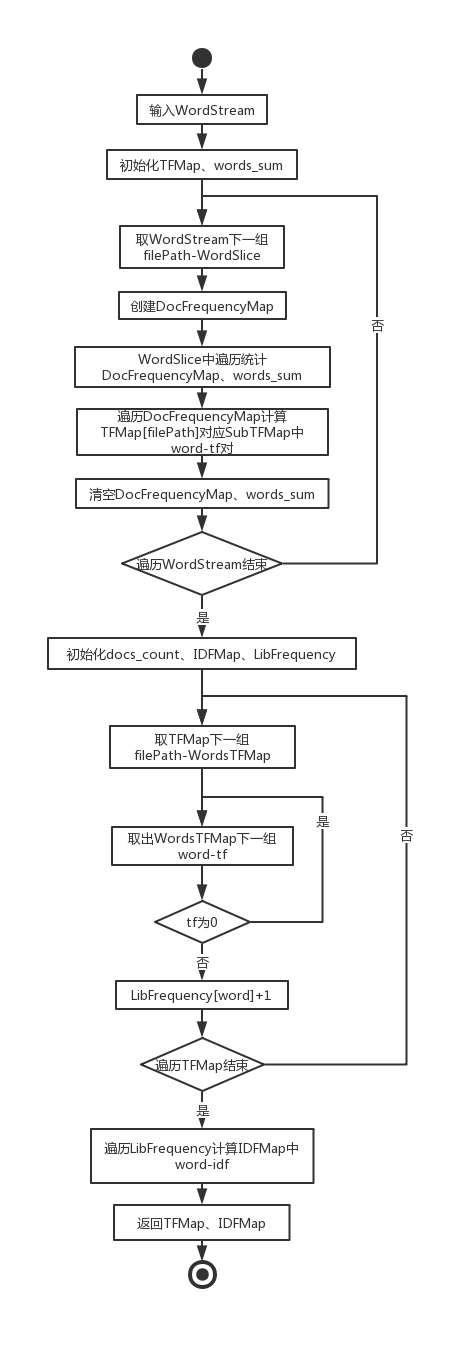
返回：无l

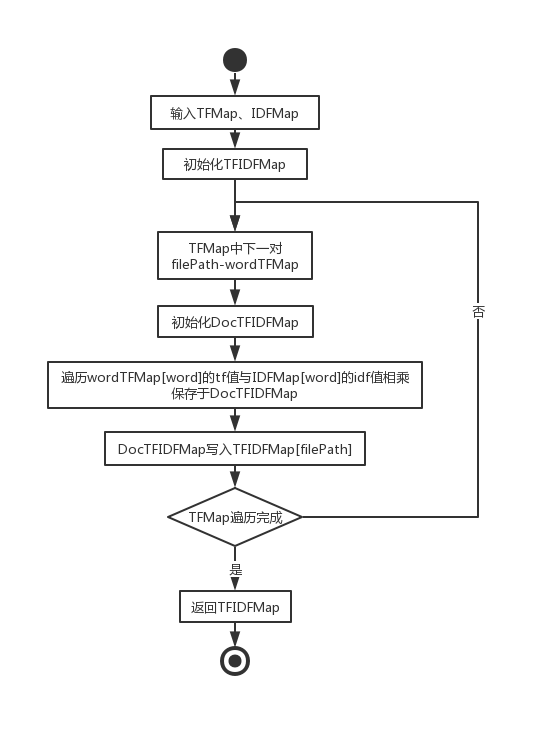
接收：无

功能：分别尝试以上各函数

## 主要实现过程

文件中的单词Term Frequency和单词自身Inverse Document Frequency计算（见下）



由上述过程结果整合计算文本中单词对应的TF-IDF（见下）

由上述过程计算文本之间的余弦相似性（见下）

