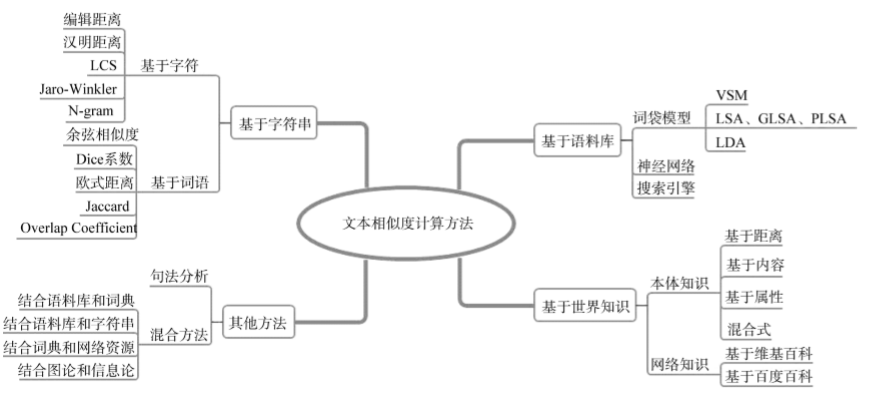
1. 目的：对国内外 文本相似度计算方法的研究现状进行系统梳理, 分析 当前各种方法的优缺点。
2. 文本相似度定义：(1) 从信息论的角度阐明相似度与文本之间的共性和差异有关, 共性越大、差异越小, 则相似度越高; 共性越小、差异越大, 则相似度越低; 相似度大的情况 是文本完全相同（文本相等程度与文本相似度）；C:\Users\mi\Documents\Tencent Files\1539381387\FileRecv\MobileFile\Image\53J1F67L]F8%GFIIL_E@%IK.png其中common(A,B)是A和B的共性信息，description(A,B)描述A，B的全部信息，公式表示相似度与文本共性成正相关。

(2)相似度一般可用[0,1]之间的实数表示, 。相似度与语义距离（理解为语义的相关性？）呈反比 关系, 语义距离越小, 相似度越高; 语义距离越大, 相似度越低。公式C:\Users\mi\Documents\Tencent Files\1539381387\FileRecv\MobileFile\Image\Y096KA[K`ADZ{OOWO2H]5TQ.png其中Dis(SA,SB)表示文本SA，SB之间的非负语义距离，阿尔法为调节因子（保证语义距离为0时公式有意义）

扩充：语义距离计算：一、通过大量语料库进行统计，将上下文信息的概率分布作为词汇语义相似度的参照依据。它是建立在两个词语语义相似当且仅当它们处于相似的上下文环境中的这一假设的基础上。它利用大规模语料库，将词语的上下文信息作为语义相似度计算的参照依据。计算量大，计算方法复杂，另外，受数据稀疏和数据噪声的干扰较大。

二、根据本体或分类关系计算词语语义距离，一般是利用一部同义词词典。一般同义词词典都是将所有的词组织在一棵或几棵树状的层次结构中。在一棵树状图中，任何结点之间有且只有一条路径，于是这条路径的长度就可以作为这两个概念的语义距离的一种度量。

1. 相似度与相关度差别（字面差别，相似属于相关）
2. 文本表示：对文本的基本处理，将半结构化或非结构化的文本转换为计算机可读的形式，文本相似度计算方法的不同本质是文本表示方法不同



1. 基于字符串：以字符串共现和重复程度为相似度衡量标准，在字面层次上的文本比较, 文本表示即为原始文本。分为基于字符的方法Character-Based和基于词语的方法Term-Based。

其中：1、一类方法单纯从字符或词语的组成考虑相似度算法, 如编辑距离、汉明距离、余 弦相似度、Dice系数、欧式距离;（只看字符/词语的重叠）

2、另一类方法还加入了字符顺序, 即字符组成和字符顺序相同是字符串相似 的必要条件, 如长公共子串(Longest Common Substring, LCS)、Jaro-Winkler；（看重叠时考虑顺序）

3、再一类方法采用集合思想, 将字符串看作由词语构成的集合, 词语共现可用集 合的交集计算, 如N-gram、Jaccard、Overlap Coefficient；（字符集合的交集；与1的区别？）

优点：简单。缺点：未考虑词语/字符的含义与关系，结果不准。

1. 基于语料库：从语料库中获取的信息计 算文本相似度。分为1、基于词袋模型的方法；2、基于神经网络的方法；3、基于搜索引擎的方法。其中1、2以待比较相似度的文档集合为语料库，3以web为语料库。

(一)：基于词袋（BOW） 基础：词语所处的上下文语境相似, 其语义则相似。（不考虑词语在文档中的顺序，将文档表示成词语的组合）。主要方法：VSM，LSA，PLSA，LDA

1、VSM：基本思想是将每篇文档表示成一个基于词频或者词频–逆文档频率(Term Frequency-Inverse Document Frequency, TF-IDF)权重的实值向量, 那么 N 篇文 档则构成n维实值空间, 其中空间的每一维都对应词项, 每一篇文档表示该空间下的一个点或者向量.。而两个文档的相似度就是两个向量的距离, 一般采用余弦相似度方法计算。

附增：TF—IDF

2、LSA：将文本从稀疏的高维词汇空间映射到低维的潜在语义空间, 在潜在语义空间计算相似性。利用奇异值分解SVD对高维的词条–文档矩阵进行处理, 去除了原始向量空间的某些“噪音”, 使数据不再稀疏（对比VSM）。本质上是通过降维提高计算准确 度, 但该算法复杂度比较高, 可移植性差。

3、PLSA：在 LSA 基础上引入主题层, 采用期 望最大化算法(EM)训练主题, 但不适用于大规模文本

4、LDA：主题模型是一个三层贝叶斯概率模型, 包含词、 主题和文档三层结构。

(二)：基于神经网络：通过神经网络模型生成词向量(Word Vector、Word Embeddings 或Distributed Representation)来计算文本相似度。词向量本质是从未标记的非结构文本中训练出的一种低维实数向量, 这种表达方式使得类似的词语在距离上更为接近,解决词袋模型中维数灾难的问题。

词向量工具：Word2Vec：

GloVe：

特点：词向量是经过训练得到的低维实数向量，维数可以人维限制，实数可以根据文本距离调整

单语义模型？

多语义模型？

匹配矩阵模型？

深层次的句子之间模型？

(三)：基于搜索引擎：基于搜索引 擎计算语义相似度的方法开始流行起来。其基本原理 是给定搜索关键词 x、y, 搜索引擎返回包含 x、y的网 页数量 f (x)、f (y)以及同时包含 x 和 y 的网页数量f (x, y)；（与我们无关）

七、基于世界知识

分类：1、基于本体知识；2、基于网络知识。（也和我们没什么关系）

1：基于本体知识：利用本体结构体系中概念之间的上下位和同位关系, 如果概念之间是语义相似的, 那么两个概念之间有且仅有一条路径。本体指指广泛的词典、叙词表、词汇表以及狭义的本体。

2：基于网络知识

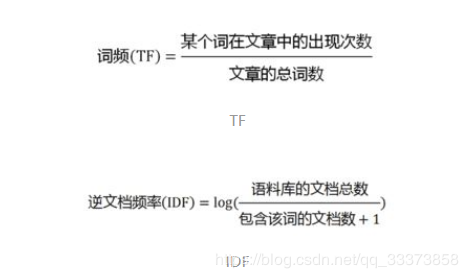
八、

（1）句法分析

（2）混合方法

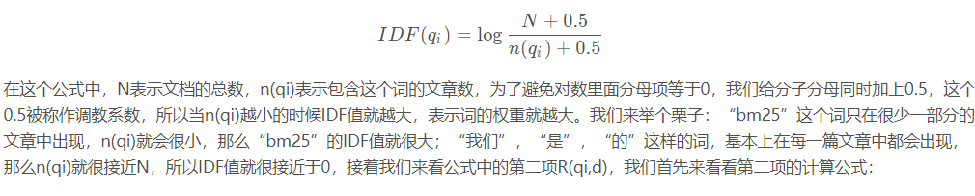
附：

1. TF-IDF

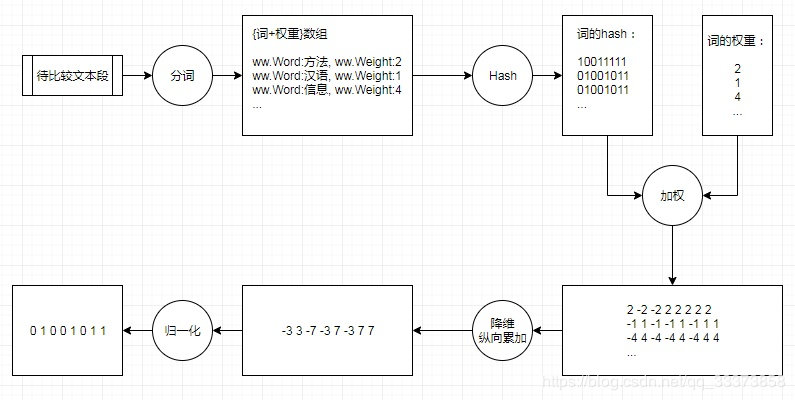
定义：一个词的权重由TF-IDF表示，其中TF表示词频，即一个词在这篇文本中出现的频率；IDF表示逆文档频率，即一个词在所有文本中出现的频率倒数。(如果按公式来说理解为出现该词的文档数/所有文档的倒数。后用余弦相似度计算。

1. BM25算法

将标准句子S1分词，每次词为qi，计算每个词与学生答案S2的相关度得分（如何计算词与句子的相关度与相似度？），最后将qi与S2的相关性得分进行加权求和，从而得到句子之间的相关性得分（不是相似性？）

用TF-IDF来计算权重Wi

1. simhash算法

主要思想：将文本分词结果从一个高位向量映射成一个0和1组成的bit指纹？然后通过比较这个二进制数字串的差异经而来表示原始问内容的差异

第一步，分词：对文本分词，同时返回当前词组在文本内容上的权重（据说所有分词工具都支持？）

第二步，计算hash：将得到的词语hash，表示为01的bit位，要求每个hash结果的位数相同

第三步、加权：根据每个词对应的权重，对hash值加权计算（1为1，0为-1），之后纵向相加，实现降维。

第四步、归一化：大于0取1，否则取0。得到simhash的指纹签名。

第五步、相似度比较：用汉明距离（将一个字符串变换成另外一个字符串需要替换的字符个数）比较，经验表示汉明距离<=3的话则认定是相似的。