VSM模型构建

2020.5

三个关键问题

- 标引项term的选择
- 权重计算(Term Weighting)：即计算每篇文档中每个term的权重
- 查询和文档的相似度计算(Similarity Computation)

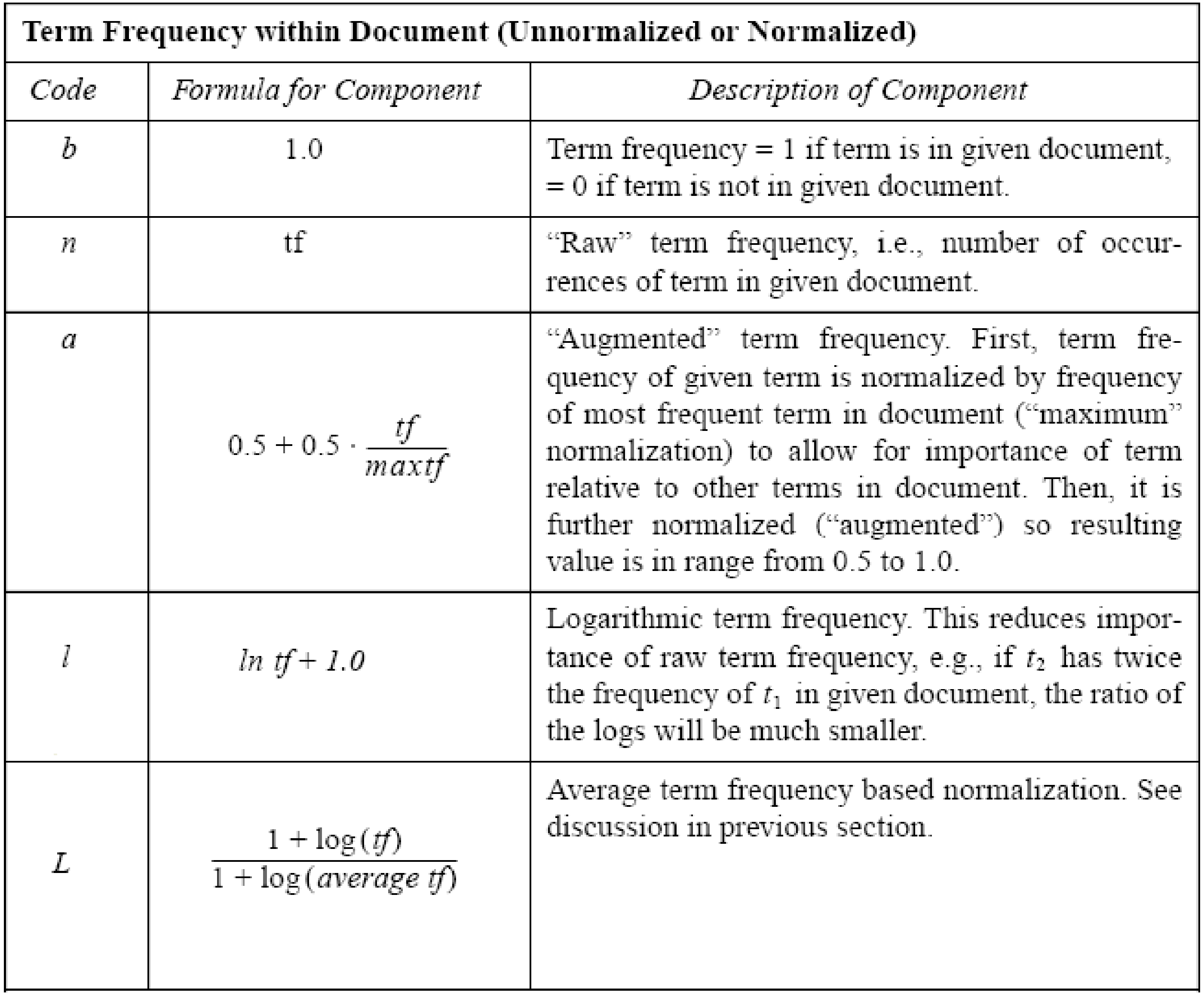
Term的选择

- term是能代表文档内容的特征
- term粒度：term可以是字、词、短语、N-gram或者某种语义单元(比如：所有同义词作为1维)，最简单的是采用全文标引(full text indexing)，即用文档中出现的所有的字或者词作为term。
- 降维：VSM中向量的维数很大(以中文词索引为例，向量维数会上10万)时，往往也同时引入了很多噪音。因此，实际应用中，会采用一些降维策略(如：去停用词、对英文进行词干还原、只选择名词作为term、将term聚成的不同类作为一个个term、选择出现次数较多的词作为term等等)

权重计算

- 文档长度因素
  - 文档长度大小不一
  - 某个term第二次出现不如第一次出现信息量大
  - 关于文档长度的两个观点：
    - 长文档具有更多的词
    - 长文档具有更多的信息
  - 因此常常需要对长文档进行惩罚，对短文档进行补偿。(不同的相似度计算方法下得到的结论不尽如此，也会采用不同的归一化方法)
  - 回转归一化(Pivoted Normalization)

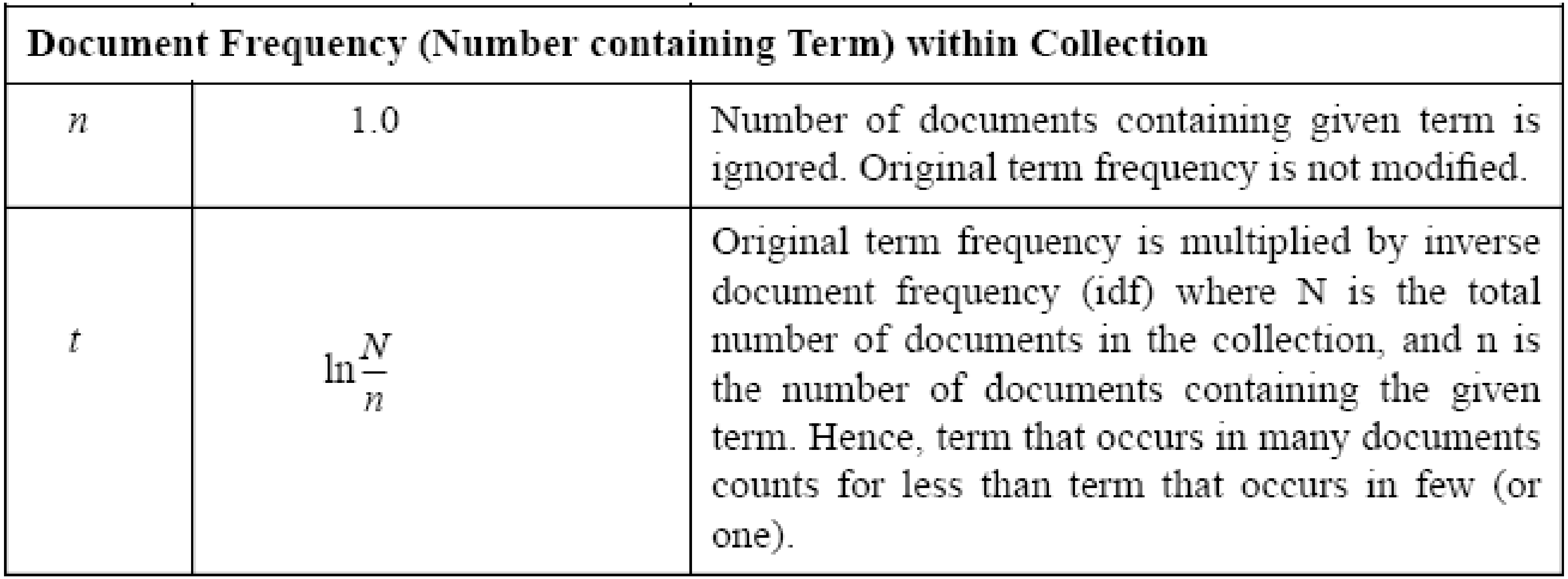
权重计算模式-TF



Term Frequency within Document (Unnormalized or Normalized)

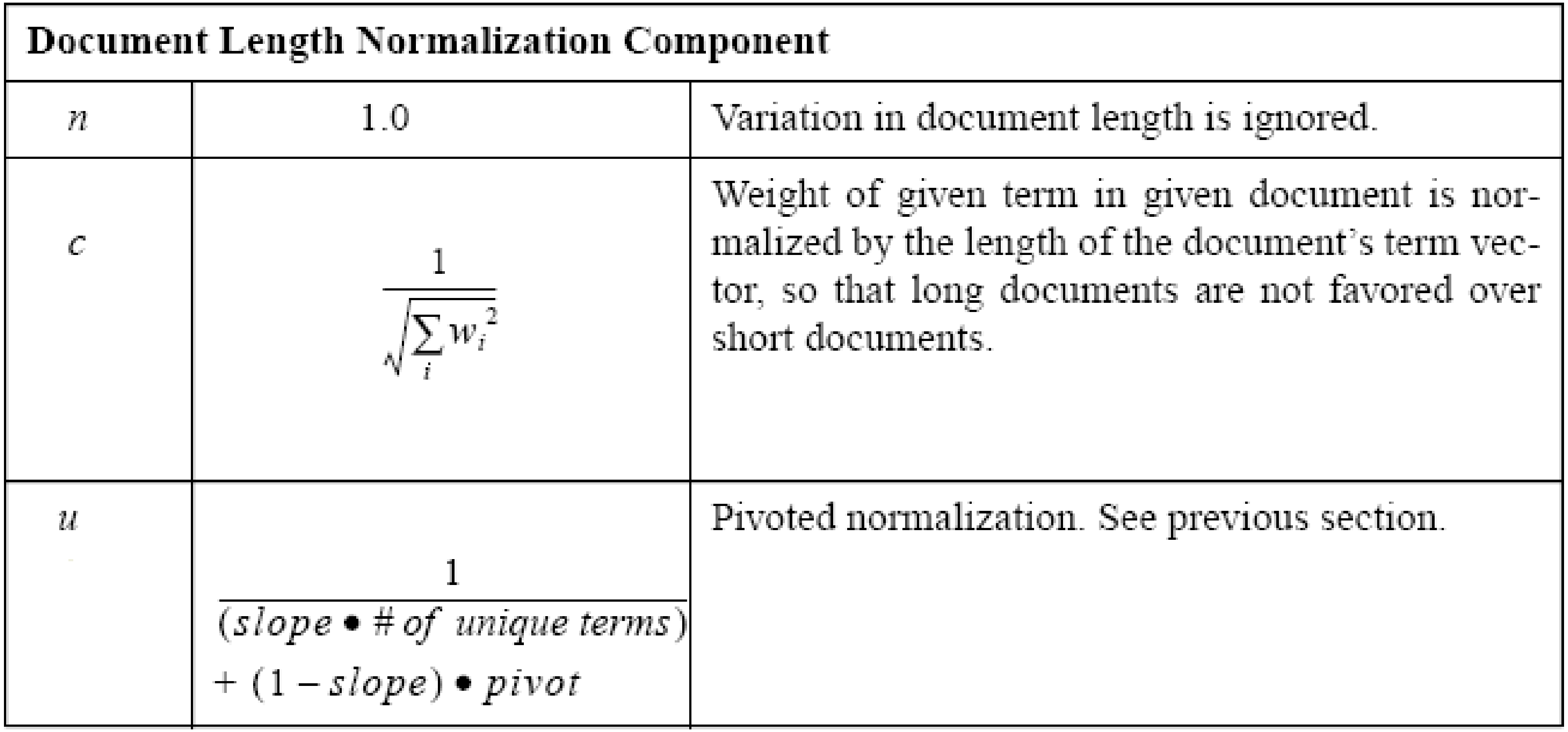
Code	*Formula for Component*	*Description of Component*
b	1.0	Term frequency = 1 if term is in given document, = 0 if term is not in given document.
n	tf	"Raw" term frequency, i.e., number of occurrences of term in given document.
a	$0.5 + 0.5 \cdot \frac{tf}{maxtf}$	"Augmented" term frequency. First, term frequency of given term is normalized by frequency of most frequent term in document ("maximum" normalization) to allow for importance of term relative to other terms in document. Then, it is further normalized ("augmented") so resulting value is in range from 0.5 to 1.0.
l	$ln\ tf + 1.0$	Logarithmic term frequency. This reduces importance of raw term frequency, e.g., if t_2 has twice the frequency of t_1 in given document, the ratio of the logs will be much smaller.
L	$\frac{1+\log(tf)}{1+\log(average\ tf)}$	Average term frequency based normalization. See discussion in previous section.

权重计算模式-IDF



Document Frequency (Number containing Term) within Collection		
n	1.0	Number of documents containing given term is ignored. Original term frequency is not modified.
t	$\ln\frac{N}{n}$	Original term frequency is multiplied by inverse document frequency (idf) where N is the total number of documents in the collection, and n is the number of documents containing the given term. Hence, term that occurs in many documents counts for less than term that occurs in few (or one).

权重计算模式-归一化

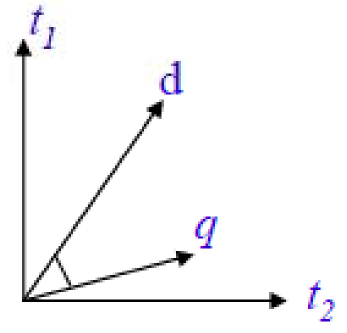


Document Length Normalization Component		
n	1.0	Variation in document length is ignored.
c	$\frac{1}{\sqrt{\sum_i w_i^2}}$	Weight of given term in given document is normalized by the length of the document's term vector, so that long documents are not favored over short documents.
u	$\frac{1}{(slope \bullet \#\ of\ \ unique\ terms) + (1 - slope) \bullet pivot}$	Pivoted normalization. See previous section.

相似度计算

内积(Dot product): $Sim(d,q) = d \bullet q = \sum_i (a_i \times b_i)$

Cosine: $$Sim(d,q) = \frac{d \bullet q}{\| d \| \times \| q \|} = \frac{\sum_i (a_i \times b_i)}{\sqrt{\sum_i a_i^2 \times \sum_i b_i^2}}$$



Dice: $$Sim(d,q) = \frac{2 \times d \bullet q}{\| d \|^2 + \| q \|^2} = \frac{2\sum_i (a_i \times b_i)}{\sum_i a_i^2 + \sum_i b_i^2}$$

Jaccard: $$Sim(d,q) = \frac{d \bullet q}{\| d \|^2 + \| q \|^2 - d \bullet q} = \frac{\sum_i (a_i * b_i)}{\sum_i a_i^2 + \sum_i b_i^2 - \sum_i (a_i * b_i)}$$

文档-标引项矩阵(Doc-Term Matrix)

- *n*篇文档，*m*个标引项构成的矩阵A_{m*n}，每列可以看成 每篇文档的向量表示，同时，每行也可以可以看成标 引项的向量表示。

$$A_{m*n} = \begin{array}{c c} & \begin{array}{cccc} d_1 & d_2 & \cdots & d_n \end{array} \\ \begin{array}{c} t_1 \\ t_2 \\ \vdots \\ t_m \end{array} & \left[\begin{array}{cccc} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{array} \right] \end{array}$$