计算文本相似度方法总结（一）

原文来自: https://www.cnblogs.com/nxf-rabbit75/p/10857008.html

方法 1：无监督，不使用额外的标注数据

- **average word vectors**：简单的对句子中的所有词向量取平均，是一种简单有效的方法，
- 缺点：**没有考虑到单词的顺序**，只对 15 个字以内的短句子比较有效，丢掉了词与词间的相关意思，无法更精细的表达句子与句子之间的关系。
- **tfidf-weighting word vectors**：指对句子中的所有词向量根据 tfidf 权重加权求和，是常用的一种计算 sentence embedding 的方法，在某些问题上表现很好，相比于简单的对所有词向量求平均，考虑到了 tfidf 权重，因此句子中更重要的词占得比重就更大。
- 缺点：没有考虑到单词的顺序
- **bag of words**：这种方法对于短文本效果很差，对于长文本效果一般，通常在科研中用来做 baseline。缺点：**1.没有考虑到单词的顺序**，2.**忽略了单词的语义信息**。
- **LDA**：计算出一片**文档或者句子的主题分布**。也常常用于文本分类任务
- 以 **smooth inverse frequency**[1]（简称 SIF)为权重，对所有词的 word vector 加权平均，最后从中减掉 principal component，得到 sentence embedding
- [1] Sanjeev Arora, et al. 2017. A Simple but Tough-to-Beat Baseline for Sentence Embeddings
- 通过 **Word Mover's Distance**[2]（简称 WMD），直接度量句子之间的相似度
- [2] Matt J. Kusner, et al. 2015. From Word Embeddings To Document Distances
- **LSI 或 LSA**：LSI 是处理相似度的，基于 SVD 分解，用于特征降维，LSI 求解出来的相似度跟 topic 相关性很强，而句子结构等信息较少。顺便说下，句子中词的顺序是不会影响 LSI 相似度结果的。

方法 2：有监督，需要额外的标注数据

- 分类任务，例如训练一个 CNN 的文本分类器[3]，取最后一个 hidden layer 的输出作为 sentence embedding，其实就是取分类器的前几层作为预训练的 encoder
- [3] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification
- sentence pair 的等价性/等义性判定[4]，这种方法的好处是不仅可以得到 sentence embedding，还可以直接学习到距离度量函数里的参数
- [4] Jonas Mueller, et al. 2016. Siamese Recurrent Architectures for Learning Sentence Similarity

方法 3：DSSM-LSTM，2016 年提出

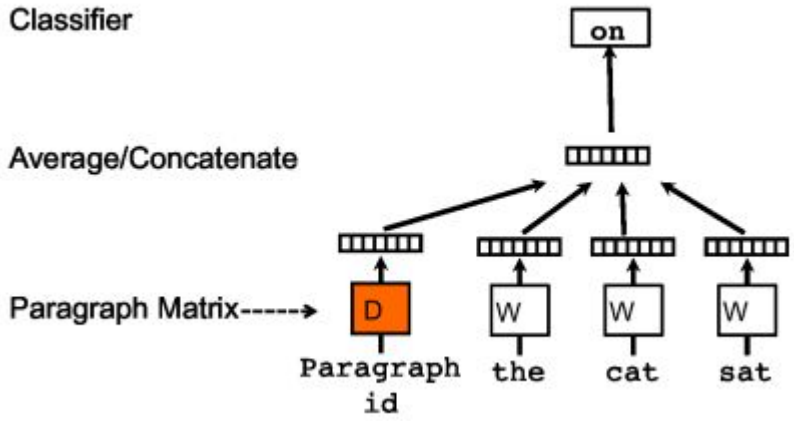
用 DSSM-LSTM 计算任意一对短文本的语义相似性，能够捕捉上下文信息。

方法 4：doc2vec（paragraph2vec, sentence embeddings），2014 年提出

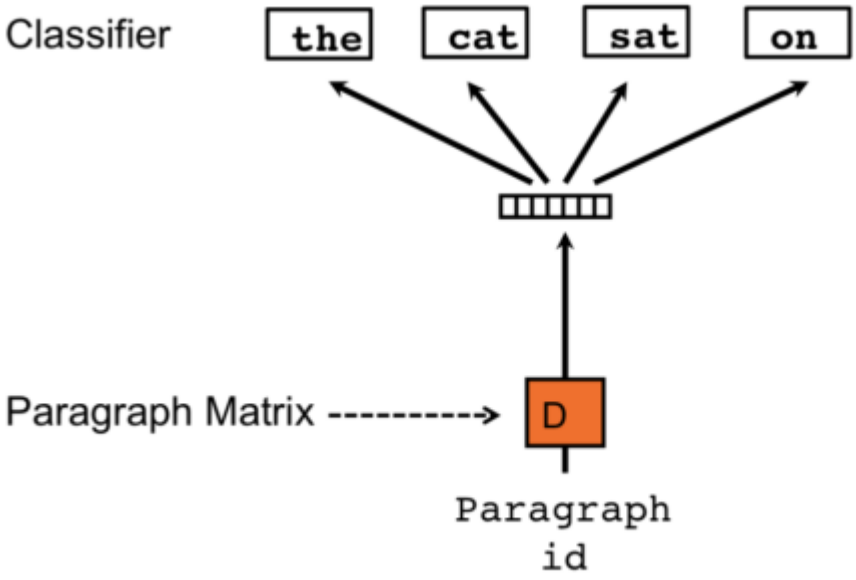
一种非监督式算法，可以获得 sentences/paragraphs/documents 的向量表达，是 word2vec 的拓展。学出来的向量可以通过计算距离来找 sentences/paragraphs/documents 之间的相似性，可以用于文本聚类，对于有标签的数据，还可以用监督学习的方法进行文本分类，例如经典的**情感分析**问题。

训练过程中新增了 paragraph id，即训练语料中每个句子都有一个唯一的 id。paragraph id 和普通的 word 一样，先是映射成一个向量，即 paragraph vector。paragraph vector 与 word vector 的维数虽一样，但是来自于两个不同的向量空间。在之后的计算里，paragraph vector 与 word vector 累加或者连接起来，作为输出层 softmax 的输入。在一个句子或者文档的训练过程中，paragraph id 保持不变，共享同一个 paragraph vector，相当于每次在预测单词的概率时，都利用了整个句子的语义。

DM(Distributed Memory，分布式内存)：DM 试图在给定前面部分的词和 paragraph 向量来预测后面单独的单词，即使文本中的语境在变化，但 paragraph 向量不会变换，并且能保存词序信息。



分布式词袋(DBOW)：利用 paragraph 来预测段落中一组随机的词.



sentence2vec 相对于 word2vec 的 skip-gram 模型，区别点为：在 sentence2vec 里，输入都是 paragraph vector，输出是该 paragraph 中随机抽样的词。

参考文献：

【1】doc2vec 原理及实践

【2】句子和文档的分布式表示

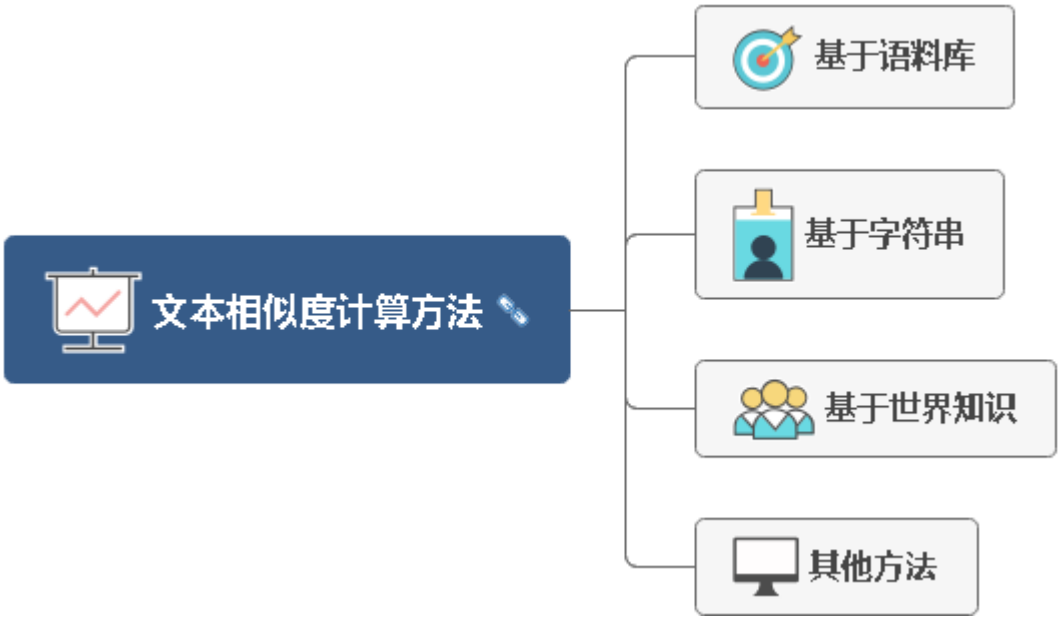
【3】基于 gensim 的 Doc2Vec 简析

【4】models.doc2vec – Doc2vec paragraph embeddings

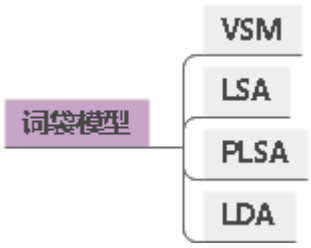
【5】如何用 word2vec 计算两个句子之间的相似度?

计算文本相似度方法总结（二）

总览

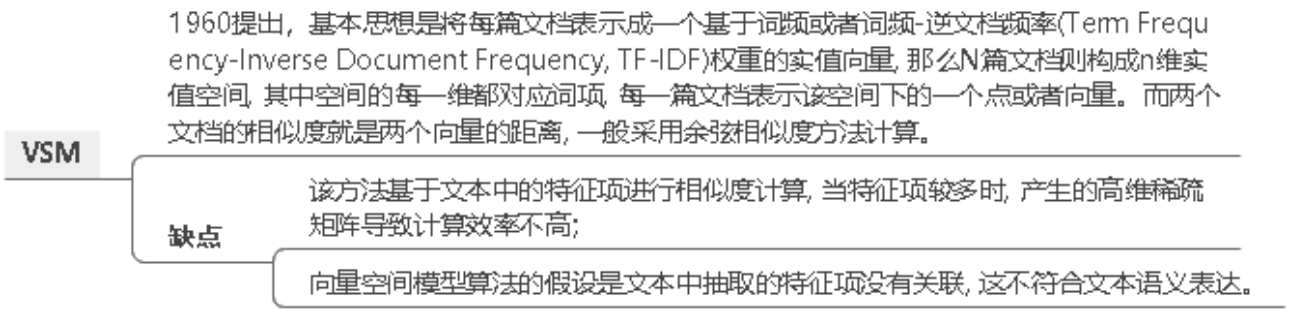


1.基于语料库

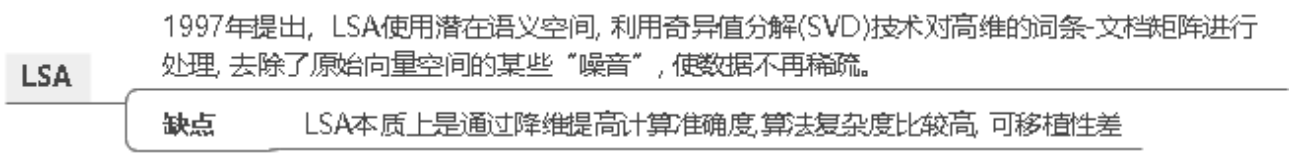


（1）词袋模型

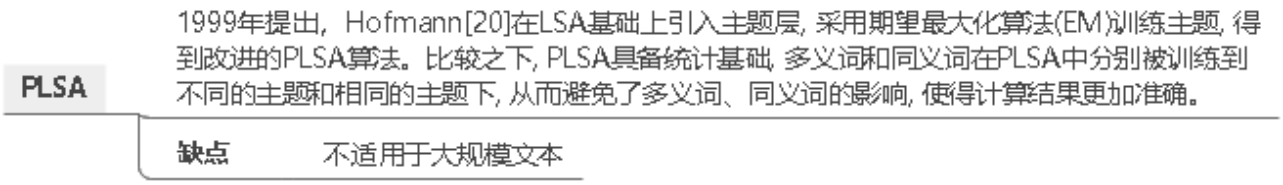
VSM



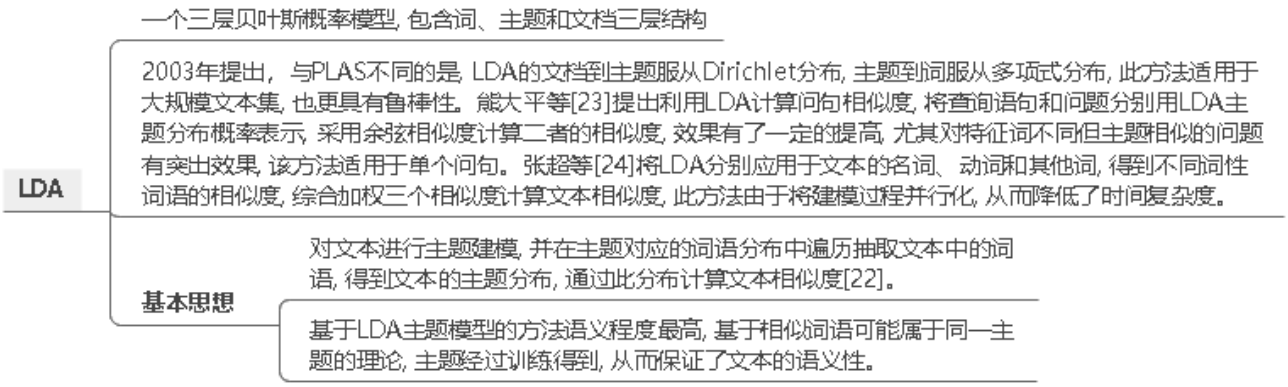
LSA



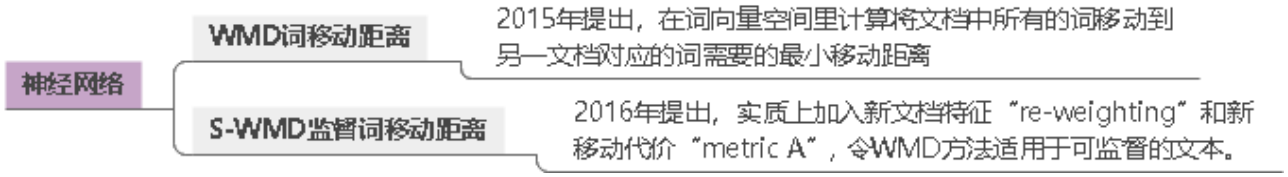
PLSA



LDA



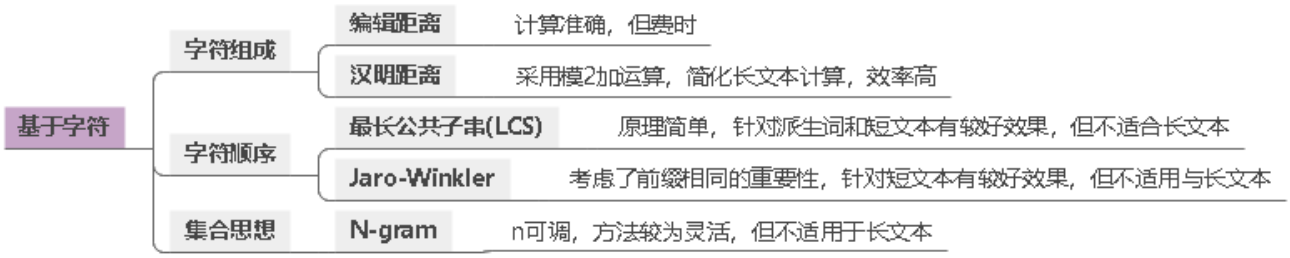
（2）神经网络



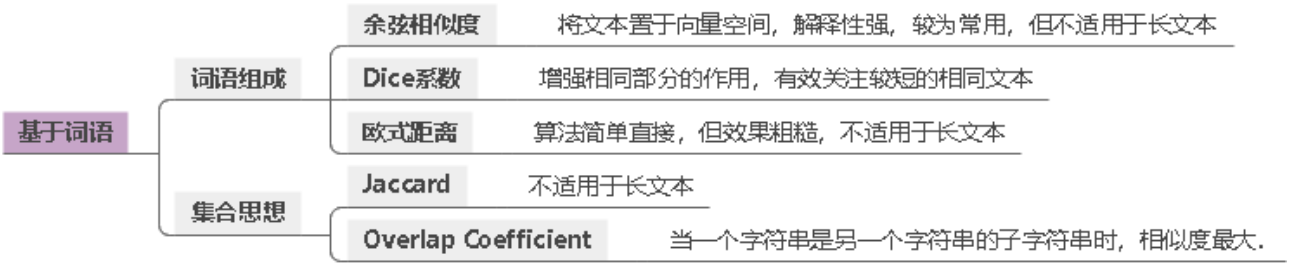
（3）搜索引擎

2.基于字符串

（1）基于字符

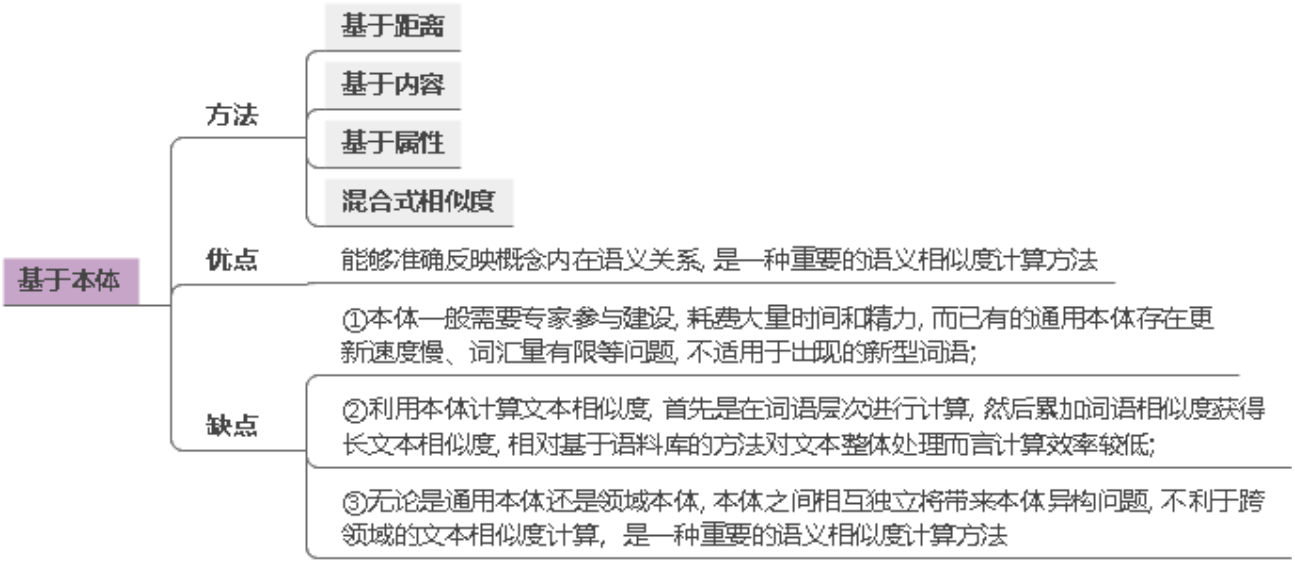


（2）基于词语

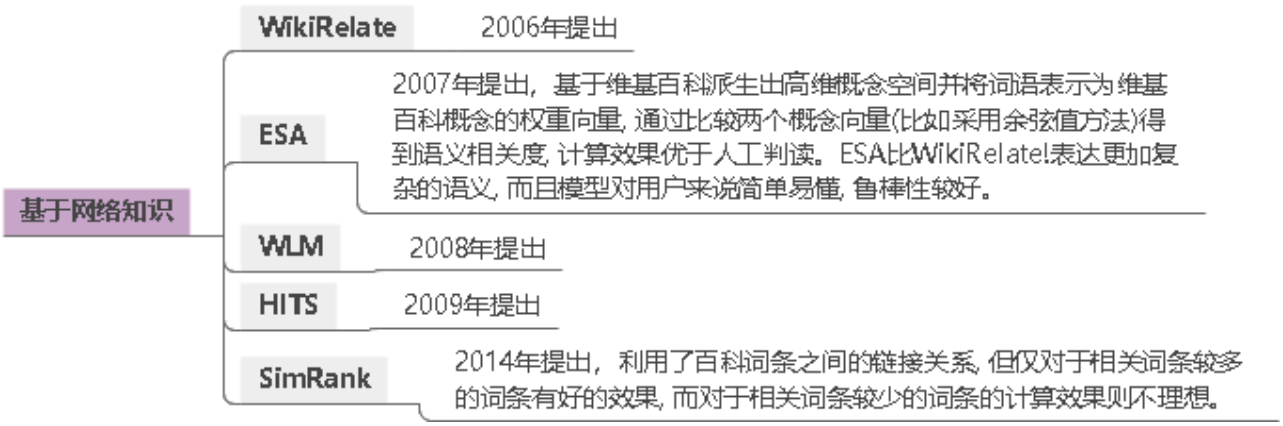


3.基于世界知识

（1）基于本体

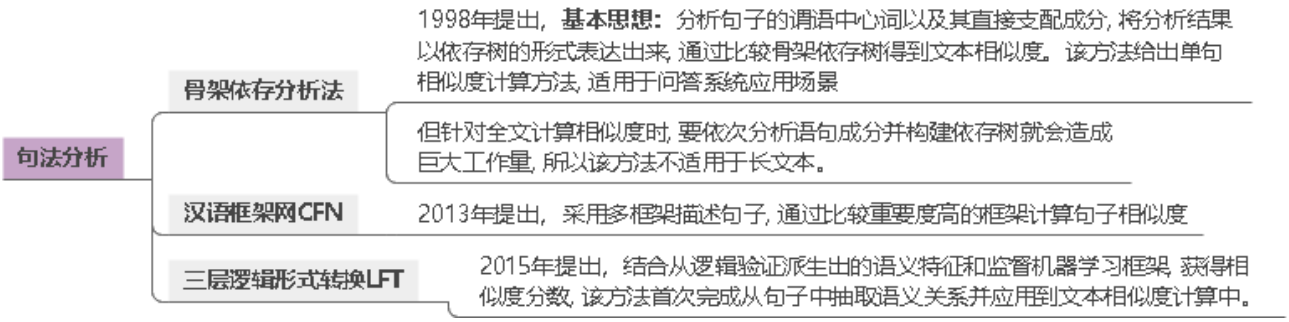


（2）基于网络知识



4.其他方法

（1）句法分析



（2）混合方式

参考文献：

【1】文本相似度计算方法研究综述 Review of Studies on Text Similarity Measures