1. **词向量**

词向量是一种单词表示形式，它将人类对语言的理解与机器的理解连接起来。词向量是文本在n维空间中的分布式表示。

**二、词向量的分类及介绍**

**1.词袋模型：one-hot、tf-idf、textrank等；**

One-hot:

词向量最直观的表示方法，以字典建立向量，词所处的位置用1表示，其余为0。此种方式的缺点是向量太稀疏造成内存消耗大，且无法表示词与词之间的语义顺序关系。

tf-idf:

tf（Term Frequency）词频：指的是某一个给定的词语在该文件中出现的次数。这个数字通常会被归一化(一般是词频除以文章总词数), 以防止它偏向长的文件。

idf（Inverse Document Frequency）逆文档频率：log（语料库的文档总数/包含该词的文档数）

特点：1. 可用某篇文章中出现次数多但在其他文章中出现次数少的词来作为该篇文章的特征词。2. 使罕见的单词更加突出并且有效地忽略了常用单词 3. 易于理解

缺点：1.考虑词的位置信息，但词的位置是有一定含义的。2. 并不能反映单词的重要程度和特征词的分布情况。

**2.主题模型：LSA（SVD）、LDA、pLSA；**

LSA（Latent Semantic Analysis）潜在语义分析：与词向量有关，是文档与所有词在该文档中出现频次的矩阵（词文档矩阵），利用SVD分解之后，左边的矩阵就是词向量。利用全局语料特征，但SVD求解计算复杂度大。

LDA（Latent Dirichlet Allocation）隐含狄利克雷分布：与词向量无关，文档集中每篇文档的主题以概率分布的形式给出。

pLSA（Probability Latent Semantic Analysis）概率潜在语义分析：太复杂了，暂无研究。

**3.基于词向量的固定表征：word2vec、fastText、GloVe等**

Word2vec:

Word2vec是Mikolov等提出模型的一个实现,可以用来快速有效地训练词向量。Word2vec包含了两种训练模型, 分别是CBOW和Skip\_gram, CBOW模型通过上下文来预测当前词, Skip\_gram模型则通过当前词来预测其上下文。Word2vec提供了两套优化方法来提高词向量的训练效率, 分别是Hierachy Softmax和Negative Sampling。

参考文献：周练.Word2vec的工作原理及应用探究[J].科技情报开发与经济,2015,25(02):145-148.

fastText:

fastText与CBOW模型比较类似，通过使用n-grams, 来缩小线性模型和深度模型之间的准确度差距, 能够取得与深度学习分类器相近的准确率, 并且在训练和评估上要比深度学习分类器快很多，目前已经有效地应用于英文标签预测和情感分析。

核心思想：将整篇文档的词及n-gram向量叠加平均得到文档向量，然后使用文档向量做softmax多分类。

GloVe:

GloVe模型是一种对 “词-词”共现矩阵X分解而得到的词向量表示方法，共现矩阵中的值X是通过滑动窗口对语料库中所有存在窗口内的目标词与上下文词的词频统计信息。它基于全局语料库、并结合上下文语境构建词向量，结合了LSA和word2vec的优点。

模型目标：进行词的向量化表示，使得向量之间尽可能多地蕴含语义和语法的信息

输入：语料库

输出：词向量

方法：首先根据语料库构建共现矩阵，然后基于共现矩阵和Glove模型学习词向量

**4.基于词向量的动态表征：elmo、GPT、bert（解决一词多义问题）**

ELMo:

ELMo的基本输入单元为句子，每个词没有固定的词向量，每个词对应的向量实际上是一个包含该词的整个句子的函数是根据词的上下文环境来动态产生当前词的词向量，常见的场景可以较好解决一词多义的问题。

参考文献及代码运行过程：https://www.jiqizhixin.com/articles/2019-04-22-3

GPT:

GPT首次无监督的预训练和有监督的微调相结合，使得模型更加符合下游任务的需求，采用单向的 Transformer 进行编码，但是相较ELMo而言可以更好地捕获长距离语言结构，同时，也降低了计算的复杂度。

BERT:

使用Transformer编码器，输入的编码向量是词向量、位置向量、句子切分向量这三个嵌入特征的单位和。利用两个监督任务进行预训练（MLM和NSP），BERT首次将无监督的预训练和有监督的微调这一模式推广到更深层的双向结构中，模型参数量众多，预训练时对硬件要求高并且消耗时间很长，一般难以自行完成预训练过程。

**三、各方法之间的比较**

Glove与Word2vec比较：

（1）Word2vec是局部语料库训练的，其特征提取是基于滑窗的；而glove的滑窗是为了构建co-occurance matrix，统计了全部语料库里在固定窗口内的词共线的频次，是基于全局语料的，可见glove需要事先统计共现概率；因此，word2vec可以进行在线学习，glove则需要统计固定语料信息；

（2）Word2vec损失函数实质上是带权重的交叉熵，权重固定；glove的损失函数是最小平方损失函数，权重可以做映射变换；

（3）Glove利用了全局信息，使其在训练时收敛更快，训练周期较word2vec较短且效果更好。

Fasttext与Word2vec比较：

（1）都可以无监督学习词向量， fastText训练词向量时会考虑subword；

（2）fastText还可以进行有监督学习进行文本分类；

（3）都利用了hierarchical softmax进行加速；

（4）Fasttext引入字符级n-gram可以处理长词，未出现过的词，以及低频词。

ELMo、GPT与BERT之间的比较：

主要体现在单双向语言模型处理上的不同：ELMo虽然是双向LSTM但是只是拼接两个方向信息，另外目标函数也只是加和两个语言模型；GPT用的是transformer的decoder，有mask的self-attention，等于只能捕获一个方向的信息；而BERT利用encoder只随机mask15%的词，利用全部上下文的信息。

四、各方法的源码

1.word2vec工具  
google开源的版本用起来速度会快很多。  
地址：<https://code.google.com/archive/p/word2vec/>  
如果无法翻墙。可看：<https://github.com/dav/word2vec>  
安装步骤：  
git clone <https://github.com/dav/word2vec>  
cd word2vec/src  
Make  
亦可尝试：./demo-word.sh 和./demo-phrases.sh

2.fastText工具

地址：https://github.com/facebookresearch/fastText

3.GloVe工具

地址： <https://github.com/stanfordnlp/GloVe>

实战操作步骤：

Linux版本：

<https://blog.csdn.net/weixin_34297300/article/details/85804332?utm_medium=distribute.pc_relevant.none-task-blog-baidujs_title-2&spm=1001.2101.3001.4242>

Windows版本：https://blog.csdn.net/keeppractice/article/details/108473693?utm\_medium=distribute.pc\_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-3.control&depth\_1-utm\_source=distribute.pc\_relevant.none-task-blog-BlogCommendFromMachineLearnPai2-3.control

4.工具gensim  
使用gensim中文处理的案例，  
by：HXY  
<https://www.zybuluo.com/hanxiaoyang/note/472184>

5.Word2vec+CNN做文本分类  
论文详见《Convolutional Neural Networks for Sentence Classification》  
<http://arxiv.org/abs/1408.5882>  
Theano完成的代码版本：  
<https://github.com/yoonkim/CNN_sentence>  
TensorFlow改写的代码版本：  
<https://github.com/dennybritz/cnn-text-classification-tf>

6.ELMo

中文训练模型：[HIT-SCIR/ELMoForManyLangs](http://link.zhihu.com/?target=https%3A//github.com/HIT-SCIR/ELMoForManyLangs)

7．BERT

官网链接：https://github.com/google-research/bert

实际操作及注释：<https://zhuanlan.zhihu.com/p/58471554>

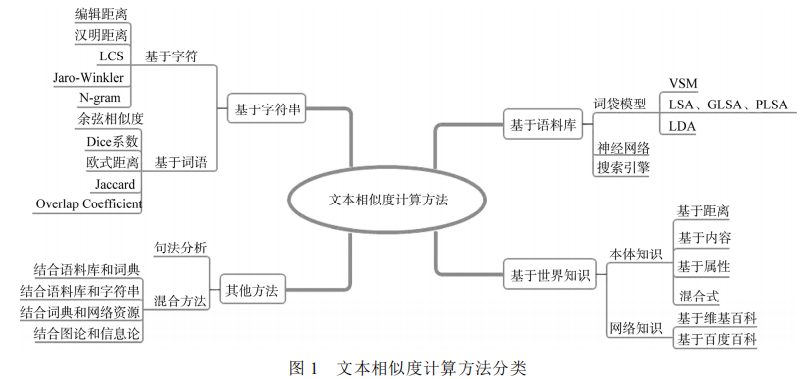
BERT的应用：

<https://github.com/jiangxinyang227/bert-for-task>

词向量相似性研究数据集：

https://github.com/CallMeJiaGu/WordSimilarityAnalogyData

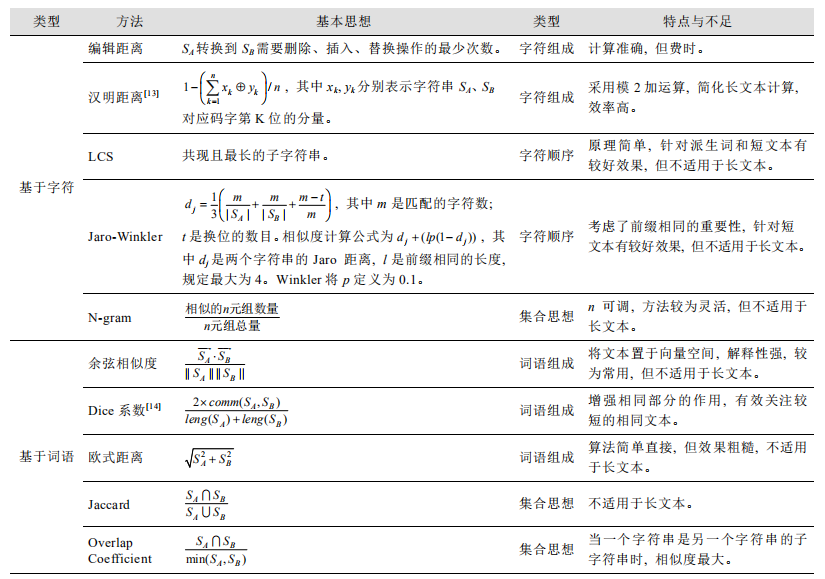
**五、文本相似度相关计算方法**

通过精读100篇文献，对文本相似度计算方法进行系统梳理,分析重点方法的基本思想、特点并总结未来发展方向。

如上图所示，将文本相似度计算方法分为4大类:基于字符串 (String-based) 的方法、基于语料库 (Corpus-based) 的方法、基于世界知识 (Knowledge-based) 的方法和其他方法。

基于字符串的方法也称作“字面相似度方法”, 其中较为典型的方法包括最长公共子串 (Longest Common Substring, LCS) 、编辑距离、Jaccard等。由于基于字符串的方法没有考虑文本的语义信息, 计算效果受到一定限制。

为解决这一问题, 学者们开始对语义相似度方法展开研究, 包括基于字符串的方法、基于语料库的方法、基于世界知识的方法和其他方法。其中其他方法又包括句法分析和混合方法, 句法分析是对句子的语法结构分析, 也属于语义分析的一种, 但其不依赖于某种语料库或世界知识, 所以被划分到其他方法。混合方法则是对几种方法的综合。

**1、基于字符串的方法**

该方法从字符串匹配度出发, 以字符串共现和重复程度为相似度的衡量标准。根据计算粒度不同, 可将方法分为基于字符 (Character-Based) 的方法和基于词语 (Term-Based) 的方法。不同方法的基本思想、类型、特点与不足如上图所示。

**2、基于语料库的方法**

从语料库中获取的信息计算文本相似度。基于语料库的方法可以分为:基于词袋模型的方法、基于神经网络的方法和基于搜索引擎的方法。前两种以待比较相似度的文档集合为语料库, 后一种以Web为语料库。

**（1）基于词袋**

根据考虑的语义程度不同, 基于词袋模型的方法主要包括向量空间模型 (Vector Space Model, VSM) 、潜在语义分析 (Latent Semantic Analysis, LSA) 、概率潜在语义分析 (Probabilistic Latent Semantic Analysis, PLSA) 和潜在狄利克雷分布 (Latent Dirichlet Allocation, LDA)。

**VSM：**将每篇文档表示成一个基于词频或者词频–逆文档频率权重的实值向量，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。

优点：原理简单；

缺点：当特征项较多时, 产生的高维稀疏矩阵导致计算效率不高;向量空间模型算法的假设是文本中抽取的特征项没有关联, 这不符合文本语义表达。

**LSA、PLSA：**

LSA算法的基本思想是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。LSA使用潜在语义空间, 利用奇异值分解技术对高维的词条–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。

PLSA算法在LSA基础上引入主题层, 采用期望最大化算法训练主题, PLSA具备统计基础, 多义词和同义词在PLSA中分别被训练到不同的主题和相同的主题下, 从而避免了多义词、同义词的影响, 使得计算结果更加准确, 但不适用于大规模文本。

**LDA：**

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。采用LDA计算文本相似性的基本思想是对文本进行主题建模, 并在主题对应的词语分布中遍历抽取文本中的词语, 得到文本的主题分布, 通过此分布计算文本相似度。与PLAS不同的是, LDA的文档到主题服从Dirichlet分布, 主题到词服从多项式分布, 此方法适用于大规模文本集, 也更具有鲁棒性。

**比较：**

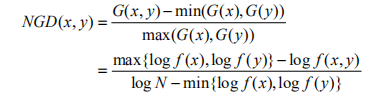
以上三类尽管都是采用词袋模型实现文本表示, 但是不同方法考虑的语义程度有所不同。基于向量空间模型的方法语义程度最低, 仅仅建立在分布假说理论基础上, 而忽略了词语之间的关联。基于LSA、PLSA的方法语义程度居中, 加入潜在语义空间概念, 解决了向量空间模型方法的稀疏矩阵问题并降低了多义词、同义词的影响。基于LDA主题模型的方法语义程度最高, 基于相似词语可能属于同一主题的理论, 主题经过训练得到, 从而保证了文本的语义性。

**（2）基于神经网络**

通过神经网络模型生成词向量是很常用的方法，如**Word2Vec、GloVe**等，词向量是经过训练得到的低维实数向量, 维数可以人为限制, 实数值可根据文本距离调整, 这种文本表示符合人理解文本的方式, 所以基于词向量判断文本相似度的效果有进一步研究空间。

**（3）基于搜索引擎**

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) ,使用如下公式计算相似度。

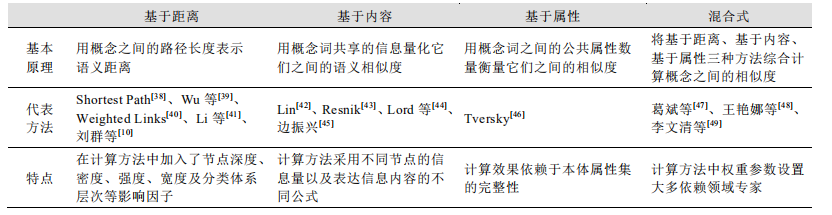


基于搜索引擎的相似度方法为相似度计算提供了丰富的语义信息, 计算结果依赖于搜索引擎的搜索效果以及对网页内容的语义分析效果, 所以精确获取返回网页数量和有效分析网页内容成为关键问题。

**3、基于世界知识的方法**

基于世界知识的方法是指利用具有规范组织体系的知识库计算文本相似度, 一般分为两种:基于本体知识和基于网络知识。

前者一般是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。网络知识中词条呈结构化并词条之间通过超链接形式展现上下位关系, 这种信息组织方式更接近计算机的理解。概念之间的路径或词条之间的链接就成为文本相似度计算的基础。

**（1）基于本体的方法**

该方法将文本表示为本体概念以及概念之间的关系, 该方法能够准确反映概念内在语义关系，其四种方法的特点如下所示：

主要缺点如下：

（1）本体一般需要专家参与建设, 耗费大量时间和精力, 而已有的通用本体存在更新速度慢、词汇量有限等问题, 不适用于出现的新型词语;

（2）利用本体计算文本相似度, 首先是在词语层次进行计算, 然后累加词语相似度获得长文本相似度, 相对基于语料库的方法对文本整体处理而言计算效率较低;

（3）无论是通用本体还是领域本体, 本体之间相互独立将带来本体异构问题, 不利于跨领域的文本相似度计算。

**（2）基于网络知识的方法**

由于本体中词语数量的限制, 有些学者开始转向基于网络知识方法的研究, 原因是后者覆盖范围广泛、富含丰富的语义信息、更新速度相对较快, 使用最多的网络知识是维基百科、百度百科。

优点：能较好地反映出词条的语义关系。

缺点:词条与词条的信息完备程度差异较大, 不能保证计算准确度;网络知识的产生方式是大众参与, 导致文本缺少一定的专业性。

**4、其他方法**

**（1）句法分析**

基于句法分析的关键是找到句子中各部分的依存关系或语义关系, 在计算相似度的同时考虑词语相似度和关系相似度, 故此方法具有更丰富的语义, 但是句子本身的复杂性为框架分析带来的难度和工作量不容小觑。

**（2）混合方法**

由于单一算法具有一定优势与不足, 所以学者综合运用两种或两种以上的方法计算文本相似度。