PathFinder算法：Path-Finder算法正是一种对复杂网络进行剪边压缩的网络优化算

法(还有聚类压缩等算法)。该算法最初由SchvaneveldtR.W.等于1988年提出,采用剪掉重要度较小的边的方法,抽取出复杂网络中实体间的主要关联关系。与其他剪边算法(如最少生成树算法等)相比,得到的网络图主要结构更加精确,在网络链接分析、检索界面设计、社会网络分析、认知结构表示、引文。网络分析、知识可视化等领域有着广泛的应用。

该算法由两个基本参数确定：r和q。r决定两个不直接相连的节点之间距离测度方法，q限定一个节点在在一个路径中可以经过的最多节点数。

PathFinder算法如何过滤非重要节点：主路径分析方法能够帮助读者快速把握一个学科领域的关键发展轨迹。主路径：从起点到终点之间累积最大遍历权重的路径。主路径上的节点应该是“重要节点”。二是主路径应该承载“最大信息量”。

Spark平台：Spark基于RDD实现了一体化、多元化的大数据处理体系，是目前最热门最高效的大数据领域的计算平台。Spark框架完美融合了SparkSQL、SparkStreaming、MLLIB、GraphX子框架，使得各子框架之间实现了数据共享和操作。

Bayes统计：贝叶斯是一种基于概率统计的学习算法。简单地说,贝叶斯理论可以概括为:为了预见未来,必须看看过去。随着统计决策理论、信息论和经验贝叶斯理论方法的创立与应用,贝叶斯方法很快显示出它的优点,成为一个活跃的研究方向。朴素贝叶斯一般分类原理就是利用类别的先验概率确定未知文本类别的后验概率。

word2vec： 深度学习word2vec是Google的一款将词表征为实数值向量的高效工具,其利用深度学习的思想,通过训练把对文本内容的处理简化为K维向量空间中的向量运算,而向量空间上的相似度可以用来表示文本语义上的相似度。在实际应用场景中,word2vec可以用于计算词的相似度、文档的相似度等,也可以用来对文档进行情感分析等。

深度学习word2vec中有CBOW和Skip-gram两种神经网络框架可用于训练并得到参数,参数迭代更新后得到的就是每个字的词向量。

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

词向量是一种分布式的低维实数向量,其基本思想是通过训练将语料型中的词语映射到N维实数向量。

可视化：数据可视化来源于科学计算可视化，科学家们不仅需要通过图形图像分析计算机算出的数据，而且需要了解计算过程中数据的变换。科学可视化指的是空间数据场的可视化，而信息可视化是指非空间数据场的可视化，比如社会信息化与网络网络应用广泛推动产生的数据。知识可视化是在科学计算可视化数据可视化信息可视化基础上发展起来的新兴研究领域应用视觉表征手段促进群体知识的传播和创新。知识可视化的实质是将人们的个体知识(暂且称为认知知识制品)以图解的手段表示出来形成能够直接作用于人的感官的知识的外在表现形式(显}生的物理知识制品)从而促进知识的传播和创新。

结构化数据：结构化数据、非结构化数据以及半结构化数据是对存储形式的一种数据类型分析，有助于企业细分行业案例，帮助存储合作伙伴更好地解决应用实施方案。结构化数据，即行数据，存储在数据库里，可以用二维表结构来逻辑表达实现的数据；简单来说就是数据库。半结构化数据，包括邮件、HTML、报表、资源库等等，典型场景如邮件系统、WEB集群、教学资源库、数据挖掘系统、档案系统等等。非结构化数据，不方便用数据库二维逻辑表来表现的数据。非结构化数据，包括视频、音频、图片、图像、文档、文本等形式。

命名实体：命名实体是文本中基本的信息元素,是正确理解文本的基础。

实体消歧：命名实体的歧义是指一个实体的指称项可以对应多个实体概念。

网络爬虫：是按照一定规则,自动地抓取万维网信息的程序或脚本,为后续的数据操作提供数据集。

垂直搜索引擎工具：我们用的是贝叶斯统计推断。

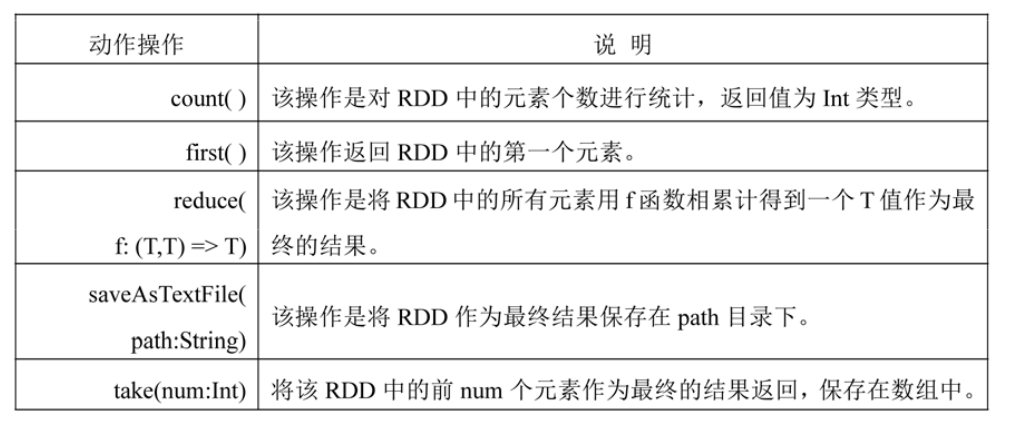
通用知识图谱：特点是不面向特定领域,可将其类比为“结构化的百科知识”。

垂直知识图谱：又称为行业知识图谱,特点是则面向特定领域,基于行业数据构建,强调知识的深度。

Spark常用转换操作举例：



Spark常用动作操作举例：

RDD（弹性分布式数据集）：该数据集具有容错功能，可以并行处理数据，它是一个高度抽象的数据结构，包含多个分区。（我记得那个慕课教程曾经举例，用在SparkSQL用RDD操作数据比在MySQL快许多？）

word分词：word分词是一个Java实现的分布式的中文分词组件，提供了多种基于词典的分词算法，并利用ngram模型来消除歧义。能准确识别英文、数字，以及日期、时间等数量词，能识别人名、地名、组织机构名等未登录词。能通过自定义配置文件来改变组件行为，能自定义用户词库、自动检测词库变化、支持大规模分布式环境，能灵活指定多种分词算法，能使用refine功能灵活控制分词结果，还能使用词频统计、词性标注、同义标注、反义标注、拼音标注等功能。提供了10种分词算法，还提供了10种文本相似度算法，同时还无缝和Lucene、Solr、ElasticSearch、Luke集成。

ngram模型：N-Gram（有时也称为N元模型）是自然语言处理中一个非常重要的概念，通常在NLP中，人们基于一定的语料库，可以利用N-Gram来预计或者评估一个句子是否合理。另外一方面，N-Gram的另外一个作用是用来评估两个字符串之间的差异程度。这是模糊匹配中常用的一种手段。（<https://blog.csdn.net/lengyuhong/article/details/6022053>）（<https://blog.csdn.net/baimafujinji/article/details/51281816>）

特征降维方法：所谓的降维就是，选出有代表性的特征词，这样就降低了维数。可以人工选择，也可以自动选择。

贝叶斯统计推断：上面提了，注意看。

SVM（支持向量机）：在机器学习中，支持向量机（英语：support vector machine，常简称为SVM，又名支持向量网络）是在分类与回归分析中分析数据的监督式学习模型与相关的学习算法。给定一组训练实例，每个训练实例被标记为属于两个类别中的一个或另一个，SVM训练算法创建一个将新的实例分配给两个类别之一的模型，使其成为非概率二元线性分类器。SVM模型是将实例表示为空间中的点，这样映射就使得单独类别的实例被尽可能宽的明显的间隔分开。然后，将新的实例映射到同一空间，并基于它们落在间隔的哪一侧来预测所属类别。

除了进行线性分类之外，SVM还可以使用所谓的核技巧有效地进行非线性分类，将其输入隐式映射到高维特征空间中。

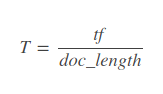
当数据未被标记时，不能进行监督式学习，需要用非监督式学习，它会尝试找出数据到簇的自然聚类，并将新数据映射到这些已形成的簇。将支持向量机改进的聚类算法被称为支持向量聚类，当数据未被标记或者仅一些数据被标记时，支持向量聚类经常在工业应用中用作分类步骤的预处理。（<https://blog.csdn.net/v_july_v/article/details/7624837>）

VSM（向量空间模型）：在该方法中，我们把一篇文章抽象成一个向量。假设向量由n个词组成，每个词的权重是kn。假设文章D，那么文章D=(k1,k2,k3,k4,k5.....kn),这是一个多维的向量。如果维数很多，将来计算起来很麻烦，我们需要降维处理。

TF/IDF：

词频(tf，Term Frequency)

一个词在一个句子中出现的次数越多，那么这个词在描述这个句子的含义方面贡献度越大，可通过下面两个式子中的一个来计算每个词的词权重：



文档频率(df)

文档频数(Document Frequency, DF)是最为简单的一种特征选择算法,它指的是在整个数据集中有多少个文本包含这个单词。在训练文本集中对每个特征计一算它的文档频次，并且根据预先设定的阑值去除那些文档频次特别低和特别高的特征。文档频次通过在训练文档数量中计算线性近似复杂度来衡量巨大的文档集，计算复杂度较低，能够适用于任何语料，因此是特征降维的常用方法。

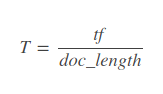
在训练文本集中对每个特征计算它的文档频数,若该项的DF 值小于某个阈值则将其删除,若其DF 值大于某个阈值也将其去掉。因为他们分别代表了“没有代表性”和“没有区分度”2 种极端的情况。DF 特征选取使稀有词要么不含有用信息,要么太少而不足以对分类产生影响,要么是噪音,所以可以删去。DF 的优点在于计算量很小,而在实际运用中却有很好的效果。缺点是稀有词可能在某一类文本中并不稀有,也可能包含着重要的判断信息,简单舍弃,可能影响分类器的精度。

文档频数最大的优势就是速度快,它的时间复杂度和文本数量成线性关系,所以非常适合于超大规模文本数据集的特征选择。不仅如此,文档频数还非常地高效,在有监督的特征选择应用中当删除90%单词的时候其性能与信息增益和x2 统计的性能还不相上下。DF 是最简单的特征项选取方法, 而且该方法的计算复杂度低, 能够胜任大规模的分类任务。

但如果某一稀有词条主要出现在某类训练集中,却能很好地反映类别的特征,而因低于某个设定的阈值而滤除掉,这样就会对分类精度有一定的影响。

逆文档频率(idf)

通常来说，如果一个词在越多的文档中出现过，那个这个词对某一个文档的贡献度应该就越小，也就是通过这个词来区分文档的区分度越小，可以用逆文档频率(idf)来度量这个概念。先定义另一个概念，文档频率df，表示包含某个词的文档的数目, N表示文档的总数量。逆文档频率计算公式如下：



TF/IDF

TF/IDF（term frequency/inverse document frequency) 的概念被公认为信息检索中最重要的发明。在搜索、文献分类和其他相关领域有广泛的应用。（<https://blog.csdn.net/quicmous/article/details/71263844>）

RDD：

### 优势[[edit](https://www.semantic-mediawiki.org/w/index.php?title=Help:使用SPARQL和RDF存储&action=edit&section=2)]

在特定的维基站点之中是否要采用某种RDF存储，取决于许多的因素，包括所采用的具体的RDF数据库。不过，我们还是可以合理地期望获得下列优势：

* 查询性能更佳  
   RDF存储在设计上旨在采用[SPARQL](https://www.semantic-mediawiki.org/wiki/SPARQL)查询语言来应答查询。与采用关系数据库的SQL查询语言相比，采用这种语言则可以更为自然地表达SMW查询。从这种意义上来说，SMW查询乃是RDF数据库系统最为典型的用例。而且，对于SMW查询来说，许多针对关系数据库查询的重要优化方法毫无用处，或者起着误导的作用。因此，可以期望，RDF存储应当可以提供极佳的查询性能。
* 额外的接口  
   支持SPARQL标准的RDF存储还允许其他应用程序针对自己的数据提出SPARQL查询，且无须借助于SMW网络前端（web frontend）。这使得我们可以在其他应用程序当中有效地利用维基站点数据。有些具备SPARQL能力的数据库还进一步支持（部分的）[OWL 2](https://www.semantic-mediawiki.org/wiki/OWL)本体语言，并提供相应的接口来查询所存储的数据（比如，借助于[OWL链接协议（OWL Link protocol）](http://www.owllink.org/" \t "_blank)）。语义网应用程序也可以利用许多的公共编程库（common programming libraries）（如[librdf](http://librdf.org/" \t "_blank)或[OWL API](http://owlapi.sourceforge.net/" \t "_blank)），从而有助于在较低的层次上将它们与其他的工具集成起来。
* 推理功能以及基于本体的数据存取能力  
   RDF Schema和 OWL之类的语义网语言，比如通过允许声明派生类（derived classes）或声明进一步的属性特性（property characteristics）（如属性的可递性<transitivity>），还为建模提供了额外的表达功能。有些具备SPARQL能力的数据库还能够评价这些查询应答功能；比如，对于基于本体的数据存取（ontology-based data access，OBDA）来说，利用语义建模构造（semantic modeling constructs），依据数据来创建“虚拟视图（virtual views）”的方法。
* 数据集成和本体重用  
   可以在RDF数据库之中存储由SMW来更新的额外数据。这样，RDF存储就可以作为数据集成和本体重用的一种平台来发挥作用。
* 计算资源的物理隔离  
   采用与MediaWiki之中不同的数据库后端，为在多个服务器之间分配任务提供了一种简便的方法。因而，尤其是可以防止复杂查询影响维基站点的基本操作，即使是它们意外地消耗了惊人数量的计算能力，也就是说，在它们搞垮了为RDF数据库提供主机托管服务的服务器的情况下。

### 缺点[[edit](https://www.semantic-mediawiki.org/w/index.php?title=Help:使用SPARQL和RDF存储&action=edit&section=3)]

然而，同时也存在着许多可能的缺点：

* 存储要求更高  
   只是在RDF数据库之中对数据进行了镜像，而不是从SQL数据库之中删除了数据。因而，需要额外的存储空间。
* 额外的维护工作  
   RDF 后端在SMW之中的安装并不难，但要运行额外的数据库管理系统，仍要花费一些精力。
* 有关性能和稳定性的疑问  
   目前，已有许多工业级的RDF数据库可用，其中有些还是免费或开源的。然而，将这些系统与SMW配合使用的经验还是有限的，因而在决定为大型SMW应用程序采用特殊的后端之前，进行一些测试还是会有所帮助。

幸运的是，无须花费太多精力，即可在基于SQL的存储后端与基于RDF的存储后端之间来回切换，因此，在试用一段时间之后，还可以再做定夺。

（<https://blog.csdn.net/u011801161/article/details/78833958>）

监督学习和非监督学习的区别

如果我们想要预测的是离散值，例如“好瓜”“坏瓜”，此类学习任务称为“分类”；如果想要预测的是连续值，例如西瓜成熟度0.95、0.37，此类学习任务称为“回归”。

学得模型后，使用其进行预测的过程称为“测试”，被预测的样本成为“测试样本”。例如在学得f后，对测试例x，可得到其预测标记y=f(x)。

我们还可以对西瓜做“聚类”，即将训练集中的西瓜分为若干组，每组称为一个“簇”；这些自动形成的簇可能对应一些潜在的概念划分，例如“浅色瓜”“深色瓜”，甚至“本地瓜”“外地瓜”。这样的学习过程有助于我们了解数据内在的规律，能为更深入地分析数据简历基础。需说明的是，在聚类学习中，“浅色瓜”“本地瓜”这样的概念我们事先是不知道的，而且学习过程中使用的训练样本通常不拥有标记信息。

根据训练数据是否拥有标记信息，学习任务可大致分为两大类：“监督学习”和“非监督学习”，分类和回归是前者的代表，而聚类是后者的代表。再解释一下，监督学习其实就是我们对输入样本经过模型训练后有明确的预期输出，非监督学习就是我们对输入样本经过模型训练后得到什么输出完全没有预期。结合西瓜的例子，监督学习就是我们知道经过模型训练后会分为好瓜或者坏瓜，而非监督学习则会将西瓜聚类为几种我们之前没有明确定义的瓜，如“浅色瓜”“外地瓜”。

各种学习函数：（<https://blog.csdn.net/tsyccnh/article/details/76769232>）

各种优化函数：（<https://blog.csdn.net/qq_21460525/article/details/70146665>）

重点关注论文：《基于Hadoop和SVM算法的中文文本分类的研究与实现\_张磊磊》、大数据文本信息分类方法研究与实现\_邹游》、《基于VSM的文本分类系统的设计和实现\_孔:振》以及所有参考文献。

训练函数和学习函数区别：训练函数和学习函数是两个不同的函数 ，网络设置中两个都有。简单的说，训练函数确定调整的大算法，是全局调整权值和阈值，考虑的是整体误差的最小； 学习函数决定调整量怎么确定，是局部调整权值和阈值，考虑的是单个神经元误差的最小。所以两者不冲突，可以一样也可以不同，就像你绕着楼跑步时，地球也在绕着太阳跑，是局部与整体的区别，既有联系又有区别，辩证统一。

训练函数是如何让误差最小的一些算法，如梯度下降，共轭梯度，这里强调算法。学习函数是指，权值和阈值的调整规则，或者称更新规则。训练函数求得权值或阈值之后，由学习函数进行调整，然后再由训练函数训练新的权值或阈值，然后再调整，反复下去。

关于CNN文本分类：

<https://www.jianshu.com/p/cb3f7ccaedf9>

深度学习模型在计算机视觉与语音识别方面取得了卓越的成就. 在 NLP 也是可以的。

卷积具有局部特征提取的功能, 所以可用 CNN 来提取句子中类似 n-gram 的关键信息。

关于RDF：

RDF(Resource Description Framework)，即资源描述框架，其本质是一个数据模型（Data Model）。它提供了一个统一的标准，用于描述实体/资源。简单来说，就是表示事物的一种方法和手段。RDF形式上表示为SPO三元组，有时候也称为一条语句（statement），知识图谱中我们也称其为一条知识。

<https://blog.csdn.net/u011801161/article/details/78833958>

卷积神经网络（Convolutional Neural Network,CNN）是一种前馈神经网络，它的人工神经元可以响应一部分覆盖范围内的周围单元，对于大型图像处理有出色表现。它包括卷积层(alternating convolutional layer)和池层(pooling layer)。

卷积神经网络是近年发展起来，并引起广泛重视的一种高效识别方法。20世纪60年代，Hubel和Wiesel在研究猫脑皮层中用于局部敏感和方向选择的神经元时发现其独特的网络结构可以有效地降低反馈神经网络的复杂性，继而提出了卷积神经网络（Convolutional Neural Networks-简称CNN）。现在，CNN已经成为众多科学领域的研究热点之一，特别是在模式分类领域，由于该网络避免了对图像的复杂前期预处理，可以直接输入原始图像，因而得到了更为广泛的应用。 K.Fukushima在1980年提出的新识别机是卷积神经网络的第一个实现网络。随后，更多的科研工作者对该网络进行了改进。其中，具有代表性的研究成果是Alexander和Taylor提出的“改进认知机”，该方法综合了各种改进方法的优点并避免了耗时的误差反向传播。

一般地，CNN的基本结构包括两层，其一为特征提取层，每个神经元的输入与前一层的局部接受域相连，并提取该局部的特征。一旦该局部特征被提取后，它与其它特征间的位置关系也随之确定下来；其二是特征映射层，网络的每个计算层由多个特征映射组成，每个特征映射是一个平面，平面上所有神经元的权值相等。特征映射结构采用影响函数核小的sigmoid函数作为卷积网络的激活函数，使得特征映射具有位移不变性。此外，由于一个映射面上的神经元共享权值，因而减少了网络自由参数的个数。卷积神经网络中的每一个卷积层都紧跟着一个用来求局部平均与二次提取的计算层，这种特有的两次特征提取结构减小了特征分辨率。

CNN主要用来识别位移、缩放及其他形式扭曲不变性的二维图形。由于CNN的特征检测层通过训练数据进行学习，所以在使用CNN时，避免了显示的特征抽取，而隐式地从训练数据中进行学习；再者由于同一特征映射面上的神经元权值相同，所以网络可以并行学习，这也是卷积网络相对于神经元彼此相连网络的一大优势。卷积神经网络以其局部权值共享的特殊结构在语音识别和图像处理方面有着独特的优越性，其布局更接近于实际 的生物神经网络，权值共享降低了网络的复杂性，特别是多维输入向量的图像可以直接输入网络这一特点避免了特征提取和分类过程中数据重建的复杂度。

主要使用一维卷积核的CNN进行文本分类（二维卷积主要用于图像处理）

步骤 1：测试文本的预处理，分词->去去除停用词->统计选择top 5000的词做为特征词

步骤 2：为每个特征词生成ID

步骤 3：将文本转化成ID序列，并将左侧补齐

步骤 4：训练集shuffle

步骤 5：Embedding Layer 将词转化为词向量

步骤 6：添加Conv1D卷积层

步骤 7：添加池化层

步骤 7：添加全连接层，loss function = binary\_crossentropy

步骤 8：输出层使用Sigmoid