文本分类算法：一项调查

摘要：近年来，复杂文档和文本的数量呈指数级增长，需要对机器学习方法有更深刻的理解，才能在许多应用中准确的对文本进行分类。许多机器学习方法在自然语言处理方面取得了卓越的成果。这些学习算法的成功依赖于它们理解复杂模型和数据中的非线性关系的能力。然而，找到合适的文本分类结构、体系结构和技术对于研究人员来说是一个挑战。本文简要讨论了文本分类算法。此概述涵盖不同的文本特征提取，降维方法，现有算法和技术以及评估方法。 最后，讨论了每种技术的局限性及其在实际问题中的应用。

关键词：文本分类；文本挖掘；文本表示；文本分类；文本分析；文档分类

1. 引言

近几十年来，文本分类问题在许多实际应用中得到了广泛的研究和解决。特别是随着自然语言处理(NLP)和文本挖掘的最新突破，许多研究人员现在对开发利用文本分类方法的应用程序很感兴趣。大多数文本分类和文档分类系统可以分解为以下四个阶段:特征提取、降维、分类器选择和评估。本文根据图 1 中所示的流水线，讨论了文本分类系统的结构和技术实现。

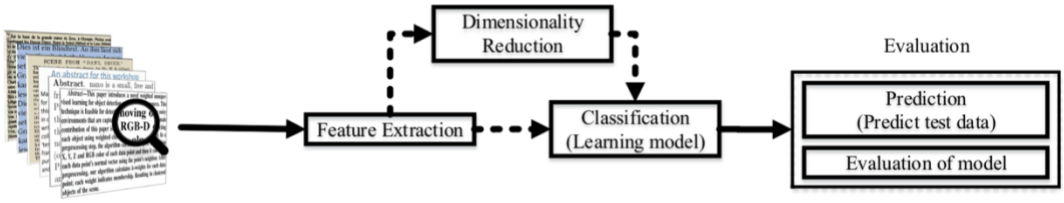

图 1. 文本分类流水线概述

初始流水线输入由一些原始文本数据集组成。通常，文本数据集包含文档中的文本序列，如$\mathrm{D} = \{X_1, X_2, \cdots, X_N\}$，其中$X_i$表示一个数据点，该数据点有 s 个句子，每个句子包含W_s个单词和I_w个字母。每个点都用一组 k 个不同离散值指标的类值标记。

然后，为了达到训练目的，我们应该创建一个被称为特征提取的结构化集合。降维的步骤是流水线上的可选部分，可以作为分类系统的一部分（例如，如果我们使用词频-逆向文件频率（TF-IDF）作为我们的特征提取，并且训练集中有我们有 200k 独特的字，计算时间是非常昂贵的，因此我们可以通过在其他维空间中引入特征空间来减少此选项。）文档分类中最重要的步骤是选择最佳的分类算

法。流水线的另一部分是评估步骤，该步骤分为两个部分(预测测试集和评估模型)。一般来说，文本分类系统包括四种不同的适用范围:

1. **文档级别**：在文档级别，算法获取完整文档的相关类别。
2. **段落级别**：在段落级别，算法获得单个段落（文档的一部分）的相关类别。
3. **句子级别**：在句子级别，获得单个句子（段落的一部分）的相关类别。
4. **子句级别**：在子句子级别，算法获得句子（句子的一部分）内的相关子表达类别)。

（I）**特征提取**：通常，文本和文档是非结构化数据集。然而，当使用数学建模作为分类器的一部分时，必须将这些非结构化文本序列转换为结构化特征空间。首先，需要清理数据以省略不必要的字符和单词。清理数据后，可以应用正式的特征提取方法。特征提取的常用技术是词频-逆向文档频率（TF-IDF），词频（TF），Word2Vec 和用词表示的全局向量（GloVe）。 在第 2 节中，我们将这些方法分类为字嵌入或加权字技术，并讨论技术实现细节。

（II）**降维**：由于文本或文档数据集通常包含许多独特的单词，数据预处理步骤可能会因时间和内存复杂性而滞后。这个问题的常见解决方案是使用廉价算法。但是，在某些数据集中，这些廉价算法的性能不如预期。为了避免性能下降，许多研究人员倾向于使用降维来减少其应用程序的时间和内存复杂性。使用降维进行预处理可能比开发廉价分类器更有效。

在第 3 节中，我们概述了最常见的降维技术，包括主成分分析（PCA），线性判别分析（LDA）和非负矩阵分解（NMF）。我们还讨论了无监督特征提取降维的新技术，如随机投影，自动编码器和 t 分布随机邻域嵌入（t-SNE）。

（III）**分类技术**：文本分类管道最重要的步骤是选择最佳分类器。如果没有对每种算法的完整概念性理解，我们就无法有效地确定文本分类应用程序的最有效模型。在第 4 节中，我们讨论了最流行的文本分类技术。首先，我们介绍了传统的文本分类方法，例如 Rocchio 分类。接下来，我们讨论基于集合的学习技术，如 boosting 和 bagging，主要用于查询学习策略和文本分析。最简单的分类算法之一是逻辑回归（LR），它已在大多数数据挖掘领域得到解决。在信息检索作为可行应用的最早历史中，朴素贝叶斯分类器（NBC）非常受欢迎。我们简要概述了朴素贝叶斯分类器，它的计算成本低廉且需要非常少的内存。

我们已经研究了非参数技术，并将其应用于 k 近邻(KNN)等分类任务中。支持向量机(SVM)是另一种使用判别分类器进行文档分类的常用技术。该技术也可

应用于生物信息学、图像、视频、人类活动分类、安全与保障等数据挖掘的各个领域。该模型还被用作许多研究人员的基准，以便与他们自己的作品进行比较以突出新颖性 和贡献。

在文档分类方面，本文还研究了基于树的分类器，如决策树和随机森林。每个基于树的算法将在单独的小节中介绍。近年来，图形分类被认为是一种分类任务，如条件随机域(CRFs)。然而，这些技术主要用于文档摘要和自动关键字提取。

近年来，深度学习方法在图像分类、自然语言处理、人脸识别等任务上取得了超过以往机器学习算法的效果。这些深度学习算法的成功依赖于它们对数据中复杂非线性关系建模的能力。

（IV）评估：文本分类流水线的最后一部分是评估。了解模型的执行方式对于文本分类方法的使用和开发至关重要。评估监督技术的方法有很多。精度计算是最简单的评估方法，但不适用于不平衡数据集。在第 5 节中，我们概述了以下文本分类算法的评估方法：F_β评分，马修斯相关系数（MCC），接收者操作特征曲线（ROC）和 ROC 曲线下面积（AUC）。

在第 6 节中，我们讨论了上述方法的局限性和缺点。并简要比较了流水线中特征提取、降维、分类技术和评价方法等步骤。在本节中，通过许多标准来比较最先进的技术，例如模型的体系结构、工作的新颖性、特征提取技术、语料库(使用的数据集/秒)、验证度量和每个工作的限制。为应用程序寻找最佳系统需要选择一种特征提取方法。此选择完全取决于应用程序的目标和数据集，因为某些特征提取技术对特定应用程序无效。例如，由于 GloVe 是在维基百科上训练的，当用于短消息服务(SMS)等短文本消息时，这种技术的性能不如 TF-IDF。此外，由于数据量小，这个模型不能像其他技术一样训练有限的数据点。下一步是分类技术，我们将简要讨论每种技术的局限性和缺点。

在第 7 节中，我们描述了文本和文档分类应用程序。文本分类是研究人员在许多领域的所面临的主要挑战。信息检索系统和搜索引擎应用程序通常使用文本分类方法。从这些应用程序扩展，文本分类还可以用于信息过滤（例如，电子邮件和文本消息垃圾邮件过滤）等应用程序 接下来，我们讨论在公共卫生和人类行为中采用文档分类。文本分类帮助的另一个领域是文档组织和知识管理。最后，我们将讨论广泛用于营销和广告的推荐系统。

2. 文本预处理

特征提取和预处理是文本分类应用的关键步骤。在本节中，我们将介绍清除文本数据集的方法，从而消除隐式噪声，并允许信息的特征化。此外，我们还讨

论了两种常用的文本特征提取方法:加权词和词嵌入技术。

2.1. 文本清理和预处理

大多数文本和文档数据集包含许多不必要的单词，如停用词、拼写错误、俚语等。在许多算法中，特别是统计和概率学习算法中，噪声和不必要的特征会对系统性能产生不利影响。在本节中，我们将简要介绍一些用于文本清理和预处理文本数据集的技术和方法。

2.1.1.标记化

标记化是一种预处理方法，它将文本流分解为单词，短语，符号或称为标记的其他有意义的元素。这一步的主要目标是调查句子中的单词。文本分类和文本挖掘都需要一个解析器来处理文档的标记化，例如：

句子：

After sleeping for four hours, he decided to sleep for another four.

在这种情况下，标记如下：

{“*After*” “*sleeping*” “*for*” “*four*” “*hours*” “*he*” “*decided*” “*to*” “*sleep*” “*for*” “*another*” “*four*”}.

2.1.2.停止词

文本和文档分类包括许多在分类算法中不具有重要意义的单词，例如{“a”，“about”，“above”，“across”，“after”，“afterwards”，“again”,…}。处理这些词的最常用技术是将它们从文本和文档中删除。

2.1.3.大写

文本和文档数据点有多种大小写形式构成句子。由于文档由许多句子组成，因此在对大型文档进行分类时，使用不同的大小写可能会带来很大的问题。处理大小写不一致的最常见方法是将每个字母都降格为小写。这种技术将文本和文档中的所有单词映射到相同的特征空间中，但是它会给一些单词(例如，“US”(美利坚合众国)到“us”(代词)的解释带来严重的问题。俚语和缩写转换器可以帮助解释这些异常。

2.1.4.俚语和缩写

俚语和缩写是在预处理步骤中处理的其他形式的文本异常。缩写是一个单词

或短语的缩写形式，它主要包含单词的首字母，比如 SVM 代表支持向量机。俚语是非正式谈话或文本中使用的语言的一个子集，有不同的含义，比如“lost the plot”，意思是他们疯了。处理这些单词的一种常见方法是将它们转换成正式语言。

2.1.5. 噪音消除

大多数文本和文档数据集包含许多不必要的字符，如标点符号和特殊字符。关键标点符号和特殊字符对于人类理解文档很重要，但对分类算法却有不利影响。

2.1.6. 拼写校正

拼写校正是一个可选的预处理步骤。错字（印刷错误的缩写）通常存在于文本和文档中，尤其是在社交媒体文本数据集中（例如，Twitter）中。许多算法，技术和方法已经在 NLP 中解决了这个问题。 研究人员可以使用许多技术和方法，许多技术和方法可供研究人员使用，包括基于哈希和上下文敏感的拼写校正技术，以及使用 Trie 距离和 Damerau-Levenshtein 距离这两种编辑距离进行拼写校正。

2.1.7. 词干提取

在 NLP 中，一个词可以以不同的形式出现（即，单数和复数名词形式），而每一种形式的语义含义相同。词干法是将不同形式的单词合并到同一特征空间的一种方法。文本词干分析是利用词缀(附加词缀)等不同的语言过程对单词进行修饰，以获得不同的单词形式。例如，单词“studying”的词根是“study”。

2.1.8. 词形还原

词形还原是一个 NLP 过程，用不同的词替换词的后缀或完全删除词的后缀以获得基本的词形式（引理）。

2.2. 语法词表示

为了解决词汇间句法和语义关系的松散问题，许多学者对文本特征提取技术进行了研究。许多研究人员提出了解决这一问题的新技术，但其中许多技术仍然有局限性。在[57]中，引入了一个模型，该模型将句法和语义知识包含在文本表示中，用于句子选择的有效性来自于技术基因组文本。另一种解决句法问题的方法是使用 n-gram 技术进行特征提取。

2.2.1 N-Gram

n-gram 技术是一组 n 个单词，在文本集中以“那个顺序”出现。这不是文本的表示，但它可以用作表示文本的特性。

BOW 是一种使用单词(1-gram)表示的文本，这些单词失去了它们的顺序(语法)。该模型非常容易获得，文本可以通过向量表示，通常是文本的可管理大小。另一方面，n-gram 是 BOW 的一个特性，用于表示使用 1-gram 的文本。使用 2-gram 和 3-gram 是很常见的。这样，提取的文本特征可以检测到比 1-gram 文本更多的信息。

一个 2-gram 例子

After sleeping for four hours, he decided to sleep for another four.

在这种情况下，标记如下：

{*“After sleeping”, “sleeping for”, “for four”, “four hours”, “four he” “he decided”, “decided to”, “to sleep”, “sleep for”, “for another”, “another four”*}.

一个 3-gram 例子

After sleeping for four hours, he decided to sleep for another four.

在这种情况下，标记如下：

{*“After sleeping for”, “sleeping for four”, “four hours he”, “hours he decided”, “he decided to”, “to sleep for”, “sleep for another”, “for another four”*}.

2.2.2. N-Gram 语法

在[58]中，讨论了 n-gram 语法，它由句法依赖或组成树中的路径而不是文本的线性结构来定义。

2.3. 加权词

加权词特征提取最基本的形式是 TF，其中每个词都映射到一个对应于该词在整个语料库中出现次数的数字。扩展 TF 结果的方法通常使用词频作为布尔或对数缩放加权。在所有加权词方法中，每个文档都被转换成一个向量(长度等于文档的长度)，其中包含该文档中单词的频率。虽然这种方法是直观的，但它会受到一种事实的限制，即语言中常用的特定单词可能主导这种表示。

2.3.1. 词袋（BoW）

词袋模型（BoW 模型）是基于特定标准（例如词频）的来自文本的选定部分

的文本文档的缩减和简化表示。

BoW 技术应用于多个领域，如计算机视觉，NLP，贝叶斯垃圾邮件过滤器，以及机器学习的文档分类和信息检索。

在 BoW 中，文本正文（例如文档或句子）被认为是一堆文字。在 BoW 过程中创建单词列表。矩阵中的这些单词不是构成句子和语法的句子，这些单词之间的语义关系在其收集和构造中被忽略。这些词通常代表句子的内容。虽然忽略了语法和出现顺序，但是会计算多样性，并可能在以后用于确定文档的焦点。

下面是 BoW 的一个例子：

文档

“As the home to UVA’s recognized undergraduate and graduate degree programs in systems engineering. In the UVA Department of Systems and Information Engineering, our students are exposed to a wide range of range”

词袋（BoW）

{“As”, “the”, “home”, “to”, “UVA’s”, “recognized”, “undergraduate”, “and”, “graduate”, “degree”, “program”, “in”, “systems”, “engineering”, “in”, “Department”, “Information”, “students”, “ ”,“are”, “exposed”, “wide”, “range”}

特征袋（BoF）

特征 ＝{1,1,1,3,2,1,2,1,2,3,1,1,1,2,1,1,1,1,1,1}

2.3.2.词袋局限性

词袋模型将词汇表中的每个单词编码为一位有效热编码向量，例如，对于大小为|Σ|的词汇表，每个单词由|Σ|维稀疏向量表示，其中单词对应的指针为 1，其他指针为 0。由于词汇量可能会达到数百万，因此词袋模型面临可扩展性挑战（例如，“这是好的”和“这是好的”具有完全相同的向量表示）。词袋的技术问题也是计算机科学和数据科学界的主要挑战。

术语频率，也称为词袋，是最简单的文本特征提取技术。 此方法基于计算每个文档中的单词数并将其分配给特征空间。

2.3.3.词频-逆向文档频率

K. Sparck Jones 提出逆向文档频率(IDF)作为一种与词频结合的方法，以减少语料库中隐含的常用词的影响。IDF 给文档中高频词或低频词赋予了更高的权

重。TF 和 IDF 的这种组合被称为词频-逆向文档频率(TF-IDF)。公式(1)给出了 TF-IDF 表示文档中某一项权重的数学表达式。

$$W(d,t) = TF(d,t) * log(\frac{N}{df(t)}) \quad (1)$$

这里 N 是文档数量，df(t)是语料库中包含术语 t 的文档数量。公式（1）中的第一项改进了回忆，而第二项改善了词嵌入的精度。虽然 TF-IDF 试图克服文件中的共同用语问题，但它仍然受到一些其他的描述限制。也就是说，TF-IDF 不能解释文档中单词之间的相似性，因为每个单词都是单独作为索引表示的。然而，随着近年来复杂模型的发展，已经提出了诸如单词嵌入的新方法，其可以结合诸如单词的相似性和部分语音标签之类的概念。

2.4. 单词嵌入

即使我们有语法词表示，但这并不意味着该模型捕获了单词的语义含义。 另一方面，词袋模型不考虑单词的语义。例如，单词“plane”、“aeroplane”、“plane”和“aircraft”经常在同一上下文中使用。然而，与这些词对应的向量在词袋模型中是正交的。这个问题对理解模型中的句子提出了一个严重的问题。单词中的另一个问题是短语中的单词顺序不会被考虑。n-gram 不能解决这个问题，因此需要为句子中的每个单词找到相似性。许多研究人员致力于字嵌入来解决这个问题。跳跃图和连续词包(CBOW)模型提出了一种简单的基于两个词向量内积的单层结构。

单词嵌入是一种特征学习技术，其中词汇表中的每个单词或短语都映射到一个 N 维实数向量。已经提出了各种单词嵌入方法来将一元图转换为机器学习算法可理解的输入。这项工作主要关注 Word2Vec、GloVe 和 FastText，这三种最常见的方法已成功地用于深度学习技术。最近，引入了一种新的单词表示技术，其中单词向量依赖于单词的上下文，被称为“语境化单词表示”或“深度语境化单词表示” 。

2.4.1. Word2Vec

T. Mikolov 等人提出了“词到向量”表示作为一种改进的词嵌入体系结构。Word2Vec 方法使用具有两个隐层的浅神经网络，连续词包(CBOW)和跳跃图模型为每个词创建高维向量。跳跃图模型包含单词 w 和上下文 c 的语料库。目的是使概率最大化：

$$arg\max_{\theta} \prod_{w \in T} \left[\prod_{c \in c(w)} p(c \mid w; \theta) \right] \quad (2)$$

其中，T 指文本，θ是$p(c|w;\theta)$的参数。

图 2 显示了一个简单的 CBOW 模型，它尝试根据以前的单词查找单词，而 Skip-gram 则尝试查找可能出现在每个单词附近的单词。输入层与输出层之间的权值表示为一个 $v \times N$ 的矩阵 w。

$$h = W^T c = W_{k,\cdot}^T := v_{wI}^T \tag{3}$$

该方法为发现文本语料库中的关系以及单词之间的相似性提供了一个非常强大的工具。例如，这种嵌入会考虑两个单词，如 “big” 和 “bigger” 在它分配的向量空间中彼此接近。

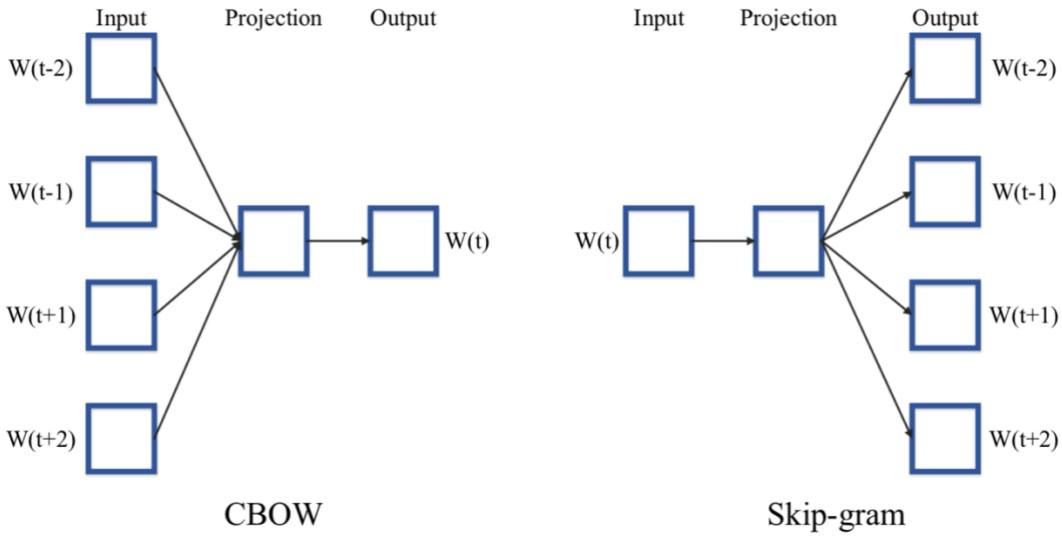

图 2. 连续词袋（CBOW）架构根据上下文预测当前单词，Skip-gram 根据给定的当前单词预测周围单词

连续词袋模型

连续词包模型由给定的目标单词的多个词表示。例如，单词“airplane” 和 “military” 作为上下文单词，而 “air-force” 作为目标单词。这包括将输入复制到隐藏层连接β次，这是上下文字的数量。因此，单词包模型主要用于将无序的单词集合表示为向量。首先要做的是创建一个词汇表，这意味着包含语料库中所有独特的单词。浅神经网络的输出将是w_i ，即任务为“给定上下文预测单词”。使用的单词数量取决于设置的窗口大小(通常大小为4-5个单词)。

连续 Skip-Gram 模型

另一个与 CBOW 非常相似的模型体系结构是连续跳跃图模型(continuous

Skip-gram model)，但是这个模型不是基于上下文来预测当前单词，而是试图基于同一句话中的另一个单词最大限度地对单词进行分类。机器学习算法采用连续词袋模型和连续 Skip-gram 模型来保持句子的句法和语义信息。

2.4.2.用词表示的全局向量（GloVe）

另一种用于文本分类的强大的单词嵌入技术是全局向量(GloVe)。该方法与 Word2Vec 方法非常相似，其中每个单词都由一个高维向量表示，并在一个巨大的语料库上基于周围的单词进行训练。许多作品中使用的预训练的单词嵌入是基于 2014 年维基百科(Wikipedia)和 Gigaword 5 上训练的 40 万个词汇作为语料库并且有 50 个用于单词表示的维度。 GloVe 还提供了其他预训练的单词矢量化，具有 100，200，300 个维度，这些维度在更大的语料库上进行训练，包括 Twitter 内容。图 3 显示了使用相同 t-SNE 技术在样本数据集上单词距离的可视化。目标函数如下:

$$f(w_i - w_j, \widetilde{w}_k) = \frac{P_{ik}}{P_{jk}} \tag{4}$$

其中，w_i 指的是单词 i 的单词向量，P_{ik} 表示单词 k 在单词 i 的上下文中出现的概率。

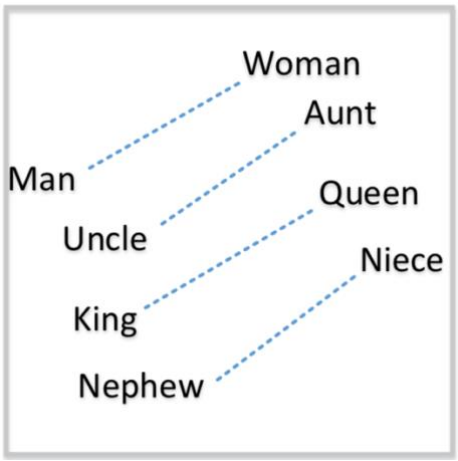

图3. GloVe: 用此表示的全局向量

2.4.3. FastText

许多其他单词嵌入表示通过为每个单词分配不同的向量来忽略单词的形态。Facebook 人工智能研究实验室推出了一项新技术来解决这个问题，该技术引入了一种名为 FastText 的新单词嵌入方法。每个单词 w 都表示为一个由 n 个字符（n-gram）组成的包。例如，给定单词“introduce”和 n = 3，FastText 将产生以

下由字符三元组组成的表示:

$$< in, int, ntr, tro, rod, odu, duc, uce, ce >$$

注意，对应于此处单词的序列<int>与单词 introduce 中 3-gram 的 “int”不同。

假设我们有一个大小为 G 的 n-gram 的字典，并给出一个单词 w，它与每个 n-gram g 的矢量表示 z_g 相关联。在这种情况下获得的评分函数是：

$$s(w,c) = \sum_{g \in g_w} z_g^T v_c \tag{5}$$

其中，$g_w \epsilon \{1,2,\ldots,G\}$。

Facebook 发布了 294 种语言的预训练单词向量，这些单词使用基于 300 维的 FastText在维基百科上进行训练。FastText使用带有默认参数的Skip-gram模型。

2.4.4. 语境化的词表示

语境化词表示是基于B. McCann等人提出的 contex2vec 技术的基础上发展起来的另一种词嵌入技术。context2vec方法使用双向长短期记忆(LSTM)。M.E. Peters等人在此技术的基础上创建了深层上下文化的单词表示技术。这种技术包含了单词表示的主要特征:(I)单词使用的复杂特征(如语法和语义)和(II)这些用法如何在不同的语言环境中变化(例如，模型的一词多义)。

这些单词嵌入技术背后的主要思想是，产生的单词向量是从双向语言模型（biLM）中学习的，该模型由前向和后向LMs组成。

前向LMs如下：

$$p(t_1,t_2,\ldots,t_N) = \prod_{k=1}^{N} p(t_k|t_1,t_2,\ldots,t_{k-1}) \tag{6}$$

后向LMs如下：

$$p(t_1,t_2,\ldots,t_N) = \prod_{k=1}^{N} p(t_k|t_{k+1},t_{k+2},\ldots,t_N) \tag{7}$$

该公式共同最大化前向和后向的对数似然性，如下所示：

$$\sum_{k=1}^{N} \begin{pmatrix} \log p(t_k|t_1,\ldots,t_{k-1};\Theta_x,\overrightarrow{\Theta}_{LSTM},\Theta_s)+ \\ \log p(t_k|t_{k+1},\ldots,t_N;\Theta_x,\overleftarrow{\Theta}_{LSTM},\Theta_s) \end{pmatrix} \tag{8}$$

其中，Θ_x是象征性表示，Θ_x指softmax层。然后，将ELMo作为所有biLM层的任务特定权重，计算如下:

$$ELMo_k^{task} = E(R_k;\Theta^{task}) = \gamma^{task} \sum_{j=0}^{L} s_j^{task} h_{k,j}^{LM} \tag{9}$$

其中，$h_{k,\mathrm{j}}^{LM}$ 计算如下：

$$h_{k,j}^{LM} = \left[\overrightarrow{h}_{k,j}^{LM}, \overleftarrow{h}_{k,j}^{LM}\right] \tag{10}$$

其中s^{task}代表softmax标准化权重，而γ^{task}是标量参数。

2.5. 局限性

虽然机器学习算法使用连续 bag-of-words 模型和连续 Skip-gram 模型来保存每个句子的句法和语义信息，但是如何保持连贯文档的完整含义仍然是机器学习的问题。

例子：

文档：{“*Maryam went to Paris on July 4th, 2018. She missed the independence day fireworks and celebrations. This day is a federal holiday in the United States commemorating the Declaration of Independence of the United States on July 4, 1776. The Continental Congress declared that the thirteen American colonies were no longer subject to the monarch of Britain and were now united, free, and independent states. She wants to stay in the country for next year and celebrate with her friends.*”}

本文档的句子级别：

S1: {“*Maryam went to Paris on July 4th, 2018.*”}

S2: {“*She missed the independence day fireworks and celebrations.*”}

S3: {“*This day is a federal holiday in the United States commemorating the Declaration of Independence of the United States on July 4, 1776.*”}

S4: {“*The Continental Congress declared that the thirteen American colonies were no longer subject to the monarch of Britain and were now united, free, and independent states.*”}

S5: {“*She has a plan for next year to stay in the country and celebrate with her friends.*”}

局限性：

图4显示了每句话的特征提取是如何失败的。图中的紫色是“This day”的简要历史介绍。而且，“This day”指的是“July 4th”。在S5中，“She”指的是S1中的“Maryam”。

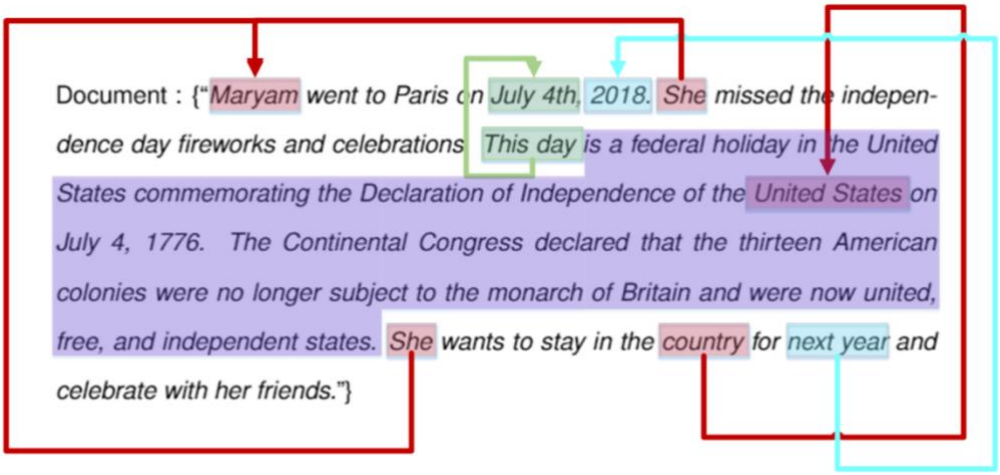

图4. 按句子级别提取文档特征的限制性。

3. 降维

3.1. 成分分析

3.1.1. 主成分分析（PCA）

主成分分析(PCA)是多变量分析和降维中最常用的方法。PCA 是一种识别数据近似所在的子空间的方法。这意味着找到不相关的新变量，并将方差最大化来“尽可能地保持可变性”。

假设给定数据集$x^{(i)}; i = 1,2, \dots, m$, 并且对于每一个$i(\mathrm{n} \leq \mathrm{m})$，$x^{(i)} \in \mathbb{R}^n$。矩阵 X 的第 j 列是向量，$x_j$是第 j 个变量的观测值。x_j的线性组合可以写成：

$$\sum_{j=1}^{m} a_j x_j = Xa \tag{11}$$

其中，a是常量a_1，a_2，$\cdots$，a_m的向量。该线性组合的方差可以表示为：

$$var(Xa) = a^T Sa \tag{12}$$

其中，S 是样本的协方差矩阵。目标是找到具有最大方差的线性组合。 这个问题转化将 $a^T Sa - \lambda(a^T a - 1)$ 最大化，其中 λ 是拉格朗日乘数。

PCA 可以作为一种预处理工具，在对数据集运行监督学习算法之前对数据集（$x^{(i)}s$作为输入）进行降维处理。PCA 作为降噪算法也是一种很有价值的工具，可以帮助避免过拟合问题。核主成分分析(KPCA)是利用核方法将线性主成分分

析推广到非线性情形的另一种降维方法。

3.1.2. 独立成分分析（ICA）

独立成分分析(ICA)是由 H. Jeanny 提出的。这项技术随后由 C. Jutten 和 J.Herault 进一步发展。ICA 是一种将观测数据表示为线性变换的统计建模方法。假设观察到 4n 个线性混合物（$x_1, x_2, \ldots, x_n$），其中独立成分为：

$$x_j = a_{j1}s_1 + a_{j2}s_2 + \ldots + a_{jn}s_n \ \forall j \tag{13}$$

向量-矩阵表示方法为：

$$X = As \tag{14}$$

用a_i表示它们，模型也可以写成：

$$X = \sum_{i=1}^{n} a_i s_i \tag{15}$$

3.2. 线性判别分析（LDA）

LDA 是一种常用的数据分类和降维技术。当类内频率不相等且性能已经在随机生成的测试数据上进行评估的情况下，LDA 尤其有用。类相关变换和类无关变换是 LDA 的两种方法，分别采用类间方差与类内方差之比，以及总方差与类内方差之比。

令$x^{(i)} \in \mathbb{R}^d$为 d 维样本，$y_i \in \{1,2,\ldots,c\}$为相关目标或输出，n 是文档数，c 为类别数，每个类的样本数量计算如下：

$$S_w = \sum_{l=1}^{c} s_l \tag{16}$$

其中，

$$S_i = \sum_{x \in w_i} (x - \mu_i)(x - \mu_i)^T, \ \mu_i = \frac{1}{N_i} \sum_{x \in w_i} x \tag{17}$$

类分散矩阵之间的一般性定义为:

$$S_B = \sum_{i=1}^{c} N_i (\mu_i - \mu)(\mu_i - \mu)^T \tag{18}$$

其中，

$$\mu = \frac{1}{N} \sum_{\forall x} x \tag{19}$$

对于可以投影到 W 矩阵中的w_i的 c-1 个投影向量:

$$W = [w_1|w_2|\dots|w_{c-1}] \tag{20}$$

$$y_i = w_i^T x \tag{21}$$

因此，投影到较低维度的 μ（均值）向量和 S 矩阵（散列矩阵）如下：

$$\widetilde{S}_w = \sum_{i=1}^{c} \sum_{y \in w_i} (y - \widetilde{\mu}_i)(y - \widetilde{\mu}_i)^T \tag{22}$$

$$\widetilde{S}_B = \sum_{i=1}^{c} (\widetilde{\mu}_i - \widetilde{\mu})(\widetilde{\mu}_i - \widetilde{\mu})^T \tag{23}$$

如果投影不是标量（c-1 维），则散列矩阵的行列式将按如下方式使用：

$$J(W) = \frac{|\widetilde{S}_B|}{|\widetilde{S}_W|} \tag{24}$$

从 Fisher 判别分析 (FDA) 中，我们可以将方程改写为：

$$J(W) = \frac{|W^T S_B W|}{|W^T S_W W|} \tag{25}$$

3.3. 非负矩阵分解（NMF）

非负矩阵因子分解(NMF)或非负矩阵近似已被证明是一种非常强大的技术，用于非常高维数据，如文本和序列分析。该技术是一种很有前途的降维方法。在本节中，将对文本和文档数据集简要介绍 NMF。给定一个$n \times m$的非负矩阵 V，可近似为：

$$V \approx WH \tag{26}$$

其中，$W = \mathbb{R}^{n \times r}$，$H = \mathbb{R}^{r \times m}$。假设$(n+m)r < nm$，则乘积 WH 可视为 V 中数据的压缩形式，则$v_i$和$h_i$为 V 和 H 的对应列。每个对应列的计算可改写为:

$$u_i \approx Wh_i \tag{27}$$

S. Tsuge 等人提出的每次迭代的计算时间可以写成:

$$\overline{H}_{ij} = H_{ij} \frac{(W^T V)_{ij}}{(W^T W H)_{ij}} \tag{28}$$

$$\overline{W}_{ij} = W_{ij} \frac{(V H^T)_{ij}}{(W H H^T)_{ij}} \tag{29}$$

因此，目标函数的局部最小值计算如下：

$$F = \sum_i \sum_j (V_{ij} - (WH)_{ij})^2 \tag{30}$$

目标函数的最大值可以被重新改写为：

$$F = \sum_i \sum_j (V_{ij} \log((WH)_{ij}) - (WH)_{ij}) \tag{31}$$

由 Kullback-Leibler 散度给出的目标函数定义如下：

$$\overline{H}_{ij} = H_{ij} \sum_k W_{kj} \frac{V_{kj}}{(WH)_{kj}} \tag{32}$$

$$\hat{W}_{ij} = W_{ij} \sum_k \frac{V_{ik}}{(WH)_{ik}} H_{jk} \tag{33}$$

$$\overline{W}_{ij} = \frac{\hat{W}_{ij}}{\sum_k \hat{W}_{kj}} \tag{34}$$

这种基于 NMF 的降维方法包含以下 5 个步骤（步骤 VI 是可选的，但通常用于信息检索：

(I) 在预处理之后提取索引项，如第 2 节中讨论的特征提取和文本清理。然后我们有 n 个具有 m 个特征的文档;
(II) 创建 n 个文档（$d \in \{d_1, d_2, \ldots, d_n\}$），其中向量$a_{ij} = \mathrm{L}_{ij} \times \mathrm{G}_i$其中 L_{ij}指文档 j 中第 i 个项的局部权重，G_i指文档 i 的全局权重;
(III) 逐一将 NMF 应用于所有文件中的所有项中;
(IV) 将训练好的文档向量投影到 r 维空间;
(V) 使用相同的变换，将测试集映射到 r 维空间;
(VI) 计算变换后的文档向量与查询向量之间的相似度。

3.4. 随机投影

随机投影是一种新的降维技术，主要应用于大容量数据集或高维特征空间。文本和文档，尤其是带有加权特征提取的文本和文档，会生成大量的特征。许多研究人员将随机投影应用到文本数据中，以用于文本挖掘、文本分类和降维。在本节中，我们将回顾一些基本的随机投影技术。如图 5 所示，展示了随机投影的概况。

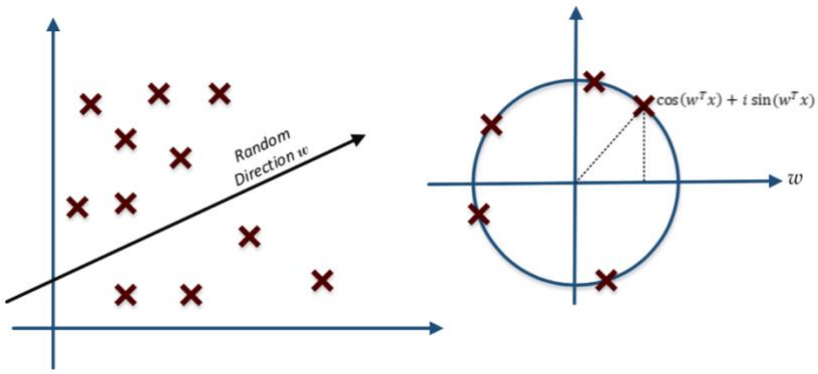

图 5. 左图表示如何生成随机方向，右图表示如何使用复数将数据集投影到新空间。

3.4.1. Random Kitchen Sinks

Random Kitchen Sinks 的关键思想是通过蒙特卡罗积分采样，将核近似作为降维的一部分。这种技术只适用于移位不变内核：

$$K(x, x') =< \phi(x), \phi(x') >\approx K(x - x') \tag{35}$$

其中移位不变内核是一个近似内核：

$$K(x - x') = z(x)z(x') \tag{36}$$

$$K(x, x') = \int_{R^D} P(w) e^{iw^T(x-x')} \tag{37}$$

其中，D 是目标样本数目，P(w)是概率分布，w 表示随机方向，$w \in \mathbb{R}^{F \times D}$，其中 F 是特征数，D 是目标。

$$K(x, x') = K(x - x') \approx \frac{1}{D} \sum_{j=1}^{D} e^{iw_j^T(x-x')} = \tag{38}$$

$$\frac{1}{D} \sum_{j=1}^{D} e^{iw_j^T x} e^{iw_j^T x'} = \frac{1}{\sqrt{D}} \sum_{j=1}^{D} e^{iw_j^T x} \frac{1}{\sqrt{D}} \sum_{j=1}^{D} e^{iw_j^T x'} \tag{39}$$

$$k(x - x') \approx \phi(x)\phi(x') \tag{40}$$

$$\phi(x) = \cos(w^T x + b_i) \tag{41}$$

其中，b_i是均匀随机变量（$b_i \in (0, \pi)$）。

3.4.2. 约翰逊-林登斯特劳斯定理

William B. Johnson 和 Joram Lindenstrauss 证明了，对于任意的 u 和$v \in n$，

$n \in \mathbb{R}^d: \exists f: \mathbb{R}^d \to \mathbb{R}^k | \varepsilon \in [0,1]$，任何欧几里得空间中的 n 个点都可以在$k = O(\frac{logn}{\varepsilon^2})$中有界。用 x=u−v 到成功概率的下限。

$$(i-\epsilon)||u-v||^2 \leq ||f(u)-f(v)||^2 \leq (i+\epsilon)||u-v||^2 \tag{42}$$

约翰逊−林登斯特劳斯定理证明：

对于来自 n 的任何 V 组数据点，其中 V∈n 和随机变量$w \in \mathbb{R}^{k \times d}$：

$$Pr[success] \geq 1-2m^2 e^{\frac{-k(\epsilon^2-\epsilon^3)}{4}} \tag{43}$$

如果我们令$k = \frac{16logn}{\varepsilon^2}$：

$$\begin{aligned} 1-2m^2 e^{\frac{-k(\epsilon^3-\epsilon^3)}{4}} \geq 1-2m^2 e^{\frac{(-\frac{8logn}{\epsilon^2})(\epsilon^2-\epsilon^3)}{4}} \\ 1-2m^2 e^{\frac{-\frac{16logn}{\epsilon^2})(\epsilon^3-\epsilon^3)}{4}} = \\ 1-2m^{4\epsilon-2} > 1-2m^{-\frac{1}{2}} > 0 \end{aligned} \tag{44}$$

定理 1 证明：

设 Ψ 是具有 k 个自由度的随机变量，对于$\varepsilon \in [0,1]$

$$Pr[(1-\epsilon)k \leq \Psi \leq (1+\epsilon)k] \geq 1-2e^{\frac{-k(\epsilon^2-\epsilon^3)}{4}} \tag{45}$$

我们从马尔科夫的不平等开始：

$$Pr[(\Psi \geq (1-\epsilon)k)] \leq \frac{E[\Psi]}{(1-\epsilon)k} \tag{46}$$

$$\begin{aligned} Pr[e^{\lambda\Psi} \geq e^{\lambda(1-\epsilon)k)}] \leq \frac{E[e^{\lambda\Psi}]}{e^{\lambda(1-\epsilon)k}} \\ E[e^{\lambda\Psi}] = (1-2\lambda)^{-\frac{k}{2}} \end{aligned} \tag{47}$$

其中λ<0.5并使用$(1-\varepsilon) \leq e^{\varepsilon-\frac{\varepsilon^2-\varepsilon^3}{2}}$ 的事实；因此，我们可以证明$Pr[(\Psi \geq (1-\varepsilon)k)] \leq \frac{E(\Psi)}{(1-\varepsilon)k}$，同理$Pr[(\Psi \leq (1+\varepsilon)k)] \leq \frac{E(\Psi)}{(1+\varepsilon)F}$ 。

$$\frac{(1+\epsilon)}{e^{\epsilon}} \leq \left(\frac{e^{\epsilon-\frac{(\epsilon^2-\epsilon^3)}{2}}}{e^{\epsilon}}\right)^{\frac{k}{2}} = e^{\frac{-k(\epsilon^3-\epsilon^3)}{4}} \tag{48}$$

$$\begin{aligned} Pr[\overline{(1-\epsilon)k \leq \Psi \leq (1+\epsilon)k}] \leq \\ Pr[(1-\epsilon)k \geq \Psi \cup \Psi \leq (1+\epsilon)k] = \\ 2e^{\frac{-k(\epsilon^3-\epsilon^3)}{4}} \end{aligned} \tag{49}$$

定理2证明：

设w是$w \in \mathbb{R}^{k \times d}$的随机变量并且k<d，对于任意$\varepsilon \in [0,1]$，x是$x \in \mathbb{R}^{d}$的数据点：

$$Pr[(1-\epsilon)||x||^2 \leq ||\frac{1}{\sqrt{k}}wx||^2 \leq (1+\epsilon)||x||^2] \geq 1-2e^{\frac{-k(\epsilon^3-\epsilon^3)}{4}} \tag{50}$$

在公式(50)中，$\frac{1}{\sqrt{k}}wx$是随机近似值并且$\hat{x} = wx$，所以我们可以重新改写公式(50)为$\Pr\left[(1-\varepsilon)\parallel x\parallel^2 \leq \parallel \frac{1}{\sqrt{k}}\hat{x}\parallel \leq (1+\varepsilon)\parallel x\parallel^2\right] \geq 1-2e^{-\frac{k(\varepsilon^3-\varepsilon^3)}{4}}$ 。

令 $\zeta_i = \frac{\hat{x}_i}{||x||} \sim N(0,1)$ 和 $\Psi = \sum_{i=1}^{k}\zeta_i^2$，因此：

$$Pr[(1-\epsilon)k \leq ||\sum_{i=0}^{k}\zeta_i||^2 \leq (1+\epsilon)k] = Pr[(1-\epsilon)k \leq ||w||^2 \leq (1+\epsilon)k] \tag{51}$$

其中，我们可以用公式(45)证明公式(51)：

$$Pr[(1-\epsilon)k \leq \Psi \leq (1+\epsilon)k] \geq 1-2e^{\frac{-k(\epsilon^3-\epsilon^3)}{4}} \tag{52}$$

3.5. 自动编码器

自动编码器是一种神经网络，它经过训练，试图将输入复制到输出中。自编码作为一种降维方法，利用了神经网络强大的可抑制性，取得了巨大成功。自动编码器的第一个版本是由D.E. Rumelhart等人在1985年推出的。主要思想是输入和输出层之间的一个隐藏层具有较少的单元，因此可用于减少特征空间的维数。特别是对于包含许多特性的文本、文档和序列，使用自动编码器可以帮助实现更快、更有效的数据处理。

3.5.1. 总体框架

如图6所示，自动编码器的输入输出层包含n个单位，其中$x = \mathbb{R}^n$，隐藏层Z包含p个单位，p < n。对于这种降维技术，最终特征空间的维数由n降至p。编码表示包括所有单词(对于词袋BoW)的表示的总和，反映每个单词的相对频率：

$$a(x) = c + \sum_{i=1}^{|x|} W_{\cdot,x_i}, \phi(x) = h(a(x)) \tag{53}$$

其中，h(.)是元素级的非线性，例如sigmoid(公式(79))。

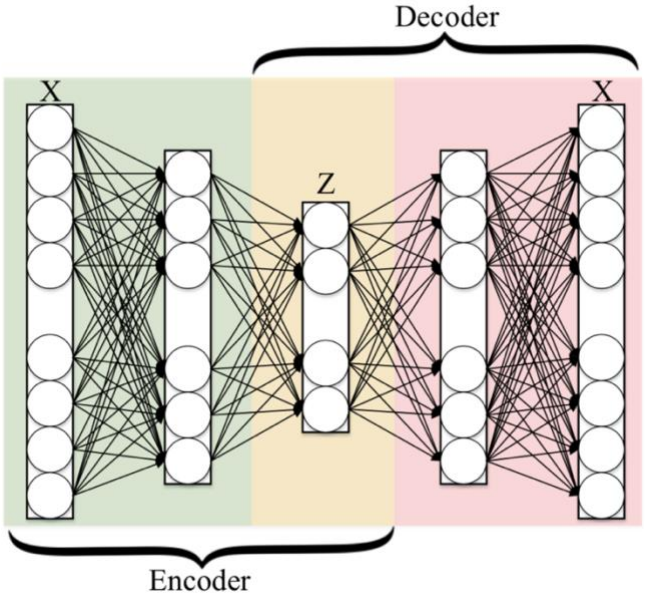

图6. 该图显示了一个简单的自动编码器的工作原理。所描述的模型包含以下层：Z是代码，两个隐层用于编码，两个隐层用于解码。

3.5.2.传统自编码器结构

一个基于卷积神经网络（CNN）的自动编码器可以被分为两个主要的步骤（编码、解码）。

$$O_m(i,j) = a\left(\sum_{d=1}^{D}\sum_{u=-2k-1}^{2k+1}\sum_{v=-2k-1}^{2k+1} F_{m_d}^{(1)}(u,v) I_d(i-u, j-v)\right) \quad \forall m = 1, \cdots, n \tag{54}$$

其中，$F \in \{F_1^{(1)}, F_2^{(1)}, \ldots, F_n^{(1)}\}$是一个卷积滤波器，在由$I = \{I_1, I_2, \ldots, I_D\}$定义的输入量中进行卷积，学习通过结合非线性函数来表示输入：

$$z_m = O_m = a(I * F_m^{(1)} + b_m^{(1)}) \quad m = 1, \cdots, m \tag{55}$$

其中，$b_m^{(1)}$是偏置，并且我们想要填充输入的零的数量满足：dim(I)= dim（decode(encode(I)）。最后，编码卷积等于：

$$\begin{aligned} O_w = O_h &= (I_w + 2(2k+1) - 2) - (2k+1) + 1 \\ &= I_w + (2k+1) - 1 \end{aligned} \tag{56}$$

解码卷积步骤产生n个特征映射$z_m = {}_{1,\cdots,n}$。重建结果$\hat{I}$是特征图量$Z = \{z_{i=1}\}^n$与该卷积滤波器量$F^{(2)}$之间的卷积的结果。

$$\tilde{I} = a(Z * F_m^{(2)} + b^{(2)}) \tag{57}$$

$$\begin{aligned} O_w = O_h = &(I_w + (2k+1) - 1) - \\ &(2k+1) + 1 = I_w = I_h \end{aligned} \tag{58}$$

其中，公式(58)显示了具有I维度的解码卷积。输入维数等于输出维数。

3.5.3. 循环自动编码器结构

递归神经网络(RNN)是将前馈神经网络自然推广到序列的一种方法。图7显示了循环自动编码器结构。一个标准的RNN通过迭代将编码计算为输出序列:

$$h_t = sigm(W^{hx}x_t + W^{hh}h_{t-1}) \tag{59}$$

$$y_t = W^{y^h}h_t \tag{60}$$

其中x为输入$(\mathrm{x}_1, x_2, \ldots, x_T)$，y为输出$(\mathrm{y}_1, y_2, \ldots, y_T)$。可以使用softmax激活函数输出多项式分布(1-K编码)。

$$p(x_{t,j} = 1 \mid x_{t-1,\ldots,x_1}) = \frac{\exp(w_j h_t)}{\sum_{j'=1}^{K} \exp(w_{j'} h_t)} \tag{61}$$

通过计算这些概率，我们可以计算得到序列x的概率为：

$$p(x) = \prod_{t=1}^{T} p(x_t \mid x_{t-1}, \ldots, x_1) \tag{62}$$

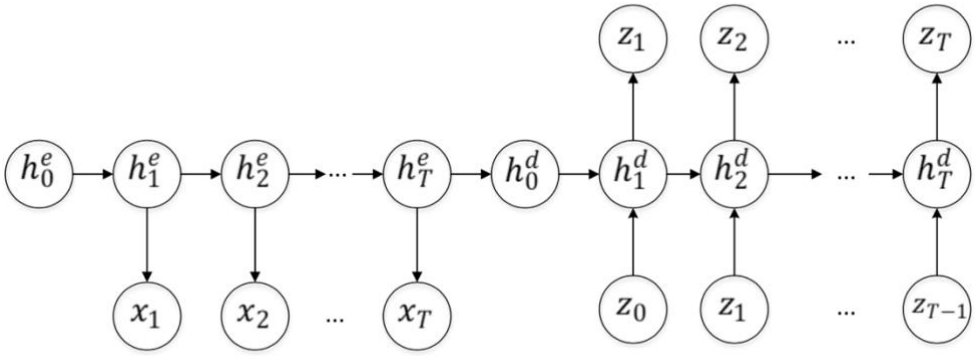

图7. 循环自动编码器结构

3.6. T分布随机领域嵌入(t-SNE)

T-SNE是一种嵌入高维数据的非线性降维方法。这种方法多用于低维特征空间的可视化，如图8所示。该方法基于G. Hinton和S. T. Roweis。SNE的工作原理是将高维欧氏距离转换为表示相似性的条件概率。条件概率$p_{j|i}$由通过以下公式计算：

$$p_{j|i} = \frac{\exp\left(-\frac{||x_i - x_j||^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{||x_i - x_k||^2}{2\sigma_i^2}\right)} \tag{63}$$

其中σ_i是以数据点x_i为中心的方差。 y_j与y_i的相似度计算如下：

$$q_{j|i} = \frac{\exp\left(-||y_i - y_j||^2\right)}{\sum_{k \neq i} \exp\left(-||y_i - y_k||^2\right)} \tag{64}$$

损失函数C如下所示：

$$C = \sum_i KL(p_i|Q_i) \tag{65}$$

其中，$KL(P_i|Q_i)$是KL散度，计算如下：

$$KL(P_i|Q_i) = \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}} \tag{66}$$

具有动量项的梯度更新如下：

$$\gamma^{(t)} = \gamma^{(t-1)} + \eta \frac{\delta C}{\delta \gamma} + \alpha(t)\left(\gamma^{(t-1)-\gamma^{(t-2)}}\right) \tag{67}$$

其中η是学习率，γ(t)是指在第t次迭代时的解，而α(t)表示在第t次迭代时的动量。现在我们可以在高维空间中重写对称SNE，在低维空间中重写联合概率分布Q，如下所示：

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{68}$$

在高维空间，p_{ij}为：

$$p_{ij} = \frac{\exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right)}{\sum_{k \neq l} \exp\left(-\frac{||x_i - x_l||^2}{2\sigma^2}\right)} \tag{69}$$

对称S的梯度如下：

$$\frac{\delta C}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j) \tag{70}$$

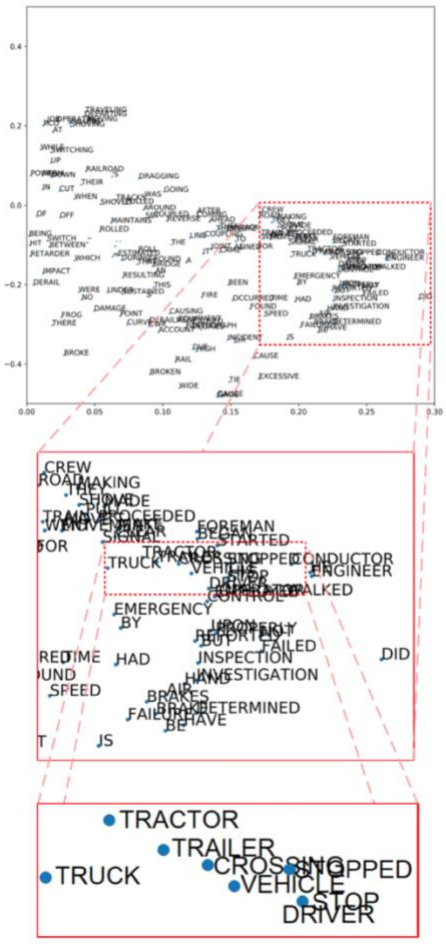

图8. 该图呈现了联邦铁路管理局(FRA)数据集的Word2vec的t分布随机邻居嵌入（t-SNE）可视化。

4. 现有的分类技术

在本节中，我们将概述现有的文本和文档分类算法。首先，我们描述了用于文本分类的Rocchio算法。然后，我们讨论了集成学习算法中的两种常用技术：Boosting和bagging。一些方法，如逻辑回归，朴素贝叶斯和k-最近邻，更传统但仍常用于科学界。支持向量机（SVM），尤其是内核SVM，也广泛用作分类技术。基于树的分类算法，例如决策树和随机森林，对于文档分类来说是快速且准确的。我们还描述了基于神经网络的算法，例如深度神经网络（DNN），CNN，RNN，深度置信网络（DBN），分层关注网络（HAN）和组合技术。

4.1. Rocchio分类

Rocchio算法最早是由 J.J.Rocchio 在1971年提出的，它是一种使用相关反馈来查询全文数据库的方法。从那时起，许多研究人员开始研究和开发这种用于文本和文档分类的技术。该分类算法对每个信息词使用TF-IDF权值，而不是布尔特

征值。使用一组训练文档，Rocchio算法为每个类构建一个原型向量。该原型是属于某一类的训练文档向量的平均向量。然后，它将每个测试文档分配给具有测试文档和每个原型向量之间的最大相似性的类。平均向量计算类c的质心（其成员的质心）：

$$\vec{\mu}(c) = \frac{1}{|D_c|} \sum_{d \in D_c} \vec{v}_d \tag{71}$$

其中D_c是D中属于c类的文档集，而$\overrightarrow{v_d}$是文档d的加权向量表示。文档d的预测标签是文档与质心之间欧氏距离最小的标签：

$$c^* = \arg\min_c \|\vec{\mu}_c - \vec{v}_d\| \tag{72}$$

质心可归一化为单位长度，如下所示：

$$\vec{\mu}_c = \frac{\sum_{d \in D_c} \vec{v}_d}{\|\sum_{d \in D_c} \vec{v}_d\|} \tag{73}$$

因此，可以得到测试文件的标签如下：

$$c_* = \arg\min_c \vec{\mu}_c \cdot \vec{v}_d \tag{74}$$

Rocchio算法的局限性

用于文本分类的Rocchio算法存在许多限制，比如用户只能使用该模型检索少量相关文档。此外，该算法的结果还需要考虑语义。

4.2. Boosting 和 Bagging

对于文档和文本数据集分类，已经成功开发出了 bagging 和 boosting等投票分类技术。然而 boosting 基于先前分类器的性能自适应地改变训练集的分布，但是 bagging 不考虑先前的分类器。

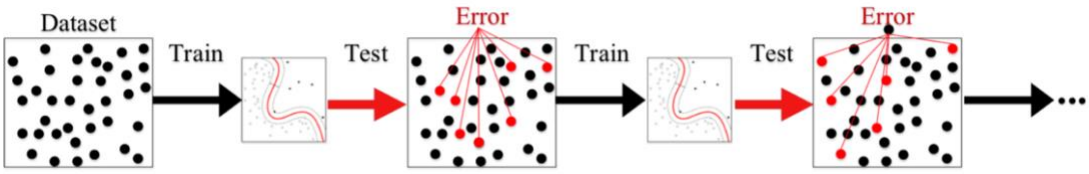

图9. 此图为boosting技术的结构

Algorithm 1 The AdaBoost method

input : training set S of size m, inducer τ, integer N

for $i = 1$ *to* N **do**
  $C_i = \tau(S')$

  $\epsilon_i = \frac{1}{m} \sum_{x_j \in S', C_i(x_j) \notin y_i} weight(x)$

  if $\epsilon_i > \frac{1}{2}$ **then**
    set S′ to a bootstrap sample from S with weight 1 for all instance and go top
  endif
  $\beta_i = \frac{\epsilon_i}{1-\epsilon_i}$
  for $x_i \in S'$ **do**
    if $C_i(x_j) = y_i$ **then**
      $weight(x_j) = weight(x_j).\beta_i$
    endif
  endfor
  Normalize weights of instances
endfor

$$C^*(x) = arg \max_{y \in Y} \sum_{i, C_i(x)=y} \log \frac{1}{\beta_i}$$

output: Classifier C^*

4.2.1. Boosting

增强算法最早由R.E.Schapire于1990年引入，作为一种提高弱学习算法性能的技术。Freund进一步发展了这项技术。

图9显示了Boosting算法如何适用于2D数据集，如图所示，我们已标记数据，然后通过多模型体系结构（集成学习）进行训练。这些发展产生了AdaBoost（Adaptive Boosting）。给定D_t和h_t，假设我们构建D_t使得$D_1(i) = \frac{1}{m}$:

$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & if y_i = h_t(x_i) \\ e^{\alpha_t} & if y_i \neq h_t(x_i) \end{cases} \\ &= \frac{D_t(i)}{Z_t} exp(-\alpha y_i h_i(x_i)) \end{aligned} \tag{75}$$

其中，Z_t指的是一个归一化系数，α_t如下：

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right) \tag{76}$$

如算法1所示，输入为大小为m的训练集S，诱导因子τ，整数N。之后该算法找到每个x_j的权重，最终输出最优分类器(C^*)。

最终的分类器公式为：

$$H_f(x) = sign\left(\sum_t \alpha_t h_t(x) \right) \tag{77}$$

4.2.2. Bagging

bagging算法是由L.Breiman于1996年提出的一种投票分类器方法。该算法由

不同的引导样本生成。一个引导程序从训练集中生成一个统一的样本。如果N个引导样本B_1，B_2，…，B_N已经生成，则我们有N个分类器（C），其中C_i是从每个自举样本B_i构建的。则我们有N个分类器（C），其中Ci是由每个引导样本B_i构建的。最后，我们的分类器C包含或生成C_1，C_2，…，C_N，其输出是其子分类器最常预测的类，并且任意断开连接。图10显示了一个简单的bagging算法，它训练了N个模型。如算法2所示，我们有经过训练的训练集S，并找到最优分类器C。

Algorithm 2 Bagging

input : training set S, inducer τ, integer N

for $i = 1$ *to* N **do**
  $S' =$ bootstrap sample from S
  $C_i = \tau(S')$
endfor

$$C^*(x) = arg \max_{y \in Y} \sum_{i, C_i = y} 1$$

output: Classifier C^*

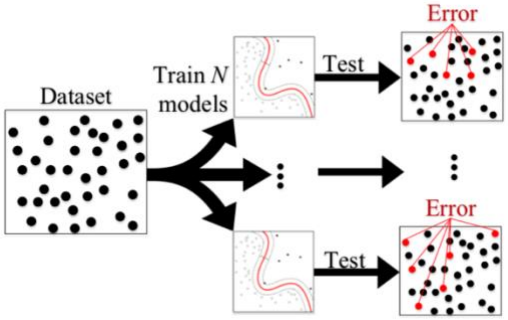

图10. 该图展现了一个Bagging技术的简单模型

4.2.3. Boosting 和 Bagging的局限性

Boosting 和 Bagging也有许多局限和缺点，例如计算复杂性和可解释性的缺失，这意味着这些模型无法发现特征重要性。

4. 3.　逻辑回归

最早的分类方法之一是逻辑回归（LR）。LR由统计学家David Cox于1958年引入和发展。LR是线性分类器，其决策边界为$\theta^T x = 0$. LR预测概率而不是类。

4. 3. 1. 基本结构

LR的目标是在给定x条件下，训练变量Y为0或1的概率。令文本数据$X \in \mathbb{R}^{n \times d}$。如果我们有二元分类问题，应该使用伯努利混合模型函数如下：

$$
\begin{aligned}
L(\theta \mid x) = p(y \mid x;\theta) = \\
&\prod_{i=1}^{n} \beta(y_i \mid sigm(x_i\theta)) = \\
&\prod_{i=1}^{n} sigm(x_i)^{y_i}(1-sigm(x_i))^{1-y_i} = \\
&\prod_{i=1}^{n} \left[\frac{1}{1+e^{-x_i\theta}}\right]^{y_i}\left[1-\frac{1}{1+e^{-x_i\theta}}\right]^{(1-y_i)}
\end{aligned}
\tag{78}
$$

其中，$x_i\theta = \theta_0 + \sum_{j=1}^{d}(x_{ij}\theta_j)$，并且$sigm(.)$是一个sigmoid函数，其定义如公式(79)所示。

$$
sigm(\eta) = \frac{1}{1-e^{-\eta}} = \frac{e^{\eta}}{1-e^{\eta}} \tag{79}
$$

4.3.2. 实例学习与LR的结合

LR模型在给定输入x_i的情况下指定二进制输出$y_i = \{0，1\}$的概率。我们可以将后验概率看作：

$$
\pi_0 = P(y_0 = +1 \mid y_i) \tag{80}
$$

其中，

$$
\frac{\pi_0}{1-\pi_0} = \frac{P(y_i \mid y_0 = +1)}{P(y_i \mid y_0 = +1)} \cdot \frac{p_0}{1-p_0} \tag{81}
$$

其中，p是似然率，它可以被重新改写为：

$$
\frac{\pi_0}{1-\pi_0} = p \cdot \frac{p_0}{1-p_0} \tag{82}
$$

$$
\log\left(\frac{\pi_0}{1-\pi_0}\right) = \log(p) + w_0 \tag{83}
$$

关于：

$$
w_0 = \log(p_0) - \log(1-p_0) \tag{84}
$$

为了遵循基于实例的学习（IBL）的基本原理，分类器应该是距离δ_i的函数。如果$\delta_i \to 0$然后$y_i = +1$，则p将大，而对于$y_i = -1$，p将小。如果$\delta_i \to \infty$，p应接近1；那么，既不赞成$y_0 = +1$也不赞成$y_0 = -1$，所以参数化函数如下：

$$
p = p(\delta) = \exp\left(y_i \cdot \frac{\alpha}{\delta}\right) \tag{85}
$$

最终，

$$
\log\left(\frac{\pi_0}{1-\pi_0}\right) = w_0 + \alpha \sum_{x_i \in N(x_0)} k(x_0, x_i).y_i \tag{86}
$$

其中，$k(x_0，x_i)$是相似性度量。

4.3.3. 多项逻辑回归

多项式(或多标记)逻辑分类使用x属于第i类的概率(如式(87)所定义)

$$p\left(y^{(i)}=1\mid x,\theta\right)=\frac{\exp\left(\theta^{(i)^T}x\right)}{\sum_{j=1}^{m}\exp\left(\theta^{(i)^T}x\right)} \tag{87}$$

其中θ（i）是对应于类i的权重向量。

对于二元分类(m = 2)，其被称为基本LR，但对于多项逻辑回归（m> 2）通常使用*softmax*函数。

归一化函数为：

$$\sum_{i=1}^{m}p\left(y^{(i)}=1\mid x,\theta\right)=1 \tag{88}$$

在作为监督学习上下文的分类任务中，θ的分量是从属于类i的训练数据D的子集计算的，其中i∈{1,...,n}。为了对θ做最大似然（ML）估计，我们需要最大化对数似然函数，如下所示：

$$\begin{aligned}\ell(\theta)&=\sum_{j=1}^{n}\log p\left(y_j=1\mid x_j,\theta\right)\\&=\sum_{j=1}^{n}\left[\sum_{i=1}^{m}y_j^{(i)}\theta^{(i)^T}x_j-\log\sum_{i=1}^{m}\exp\left(\theta^{(i)^T}x_j\right)\right]\end{aligned} \tag{89}$$

采用最大后验（MAP）估计如下：

$$\hat{\theta}_{MAP}=arg\max_{\theta}L(\theta)=arg\max_{\theta}\left[\ell(\theta)+\log p(\theta)\right] \tag{90}$$

4.3.4. 逻辑回归局限性

逻辑回归分类器在预测分类结果方面效果良好。然而，这种预测要求每个数据点是独立的，它试图基于一组独立变量来预测结果。

4.4. 朴素贝叶斯分类器

自20世纪50年代以来，朴素贝叶斯文本分类已广泛用于文档分类任务。朴素贝叶斯分类器方法在理论上基于贝叶斯定理，由Thomas Bayes在1701-1761之间制定。近年来，这一技术在信息检索领域得到了广泛的研究。该技术是一种生成模型，是最传统的文本分类方法。我们从最基本的NBC版本开始，它是使用TF（单词包）开发的，这是一种特征提取技术，用于计算文档中的单词数。

4.4.1. 朴素贝叶斯分类器的高级描述

如果文档的数量（n）符合k个类别，其中$k\in\{c_1,c_2,\dots,c_k\}$，则作为输出的预测的类是$c\in C$。朴素贝叶斯算法可以描述如下：

$$P(c\mid d)=\frac{P(d\mid c)P(c)}{P(d)} \tag{91}$$

其中，d是文档，c表示类。

$$C_{MAP} = \arg\max_{c \in C} P(d \mid c)P(c) = \arg\max_{c \in C} P(x_1, x_2, \ldots, x_n \mid c)p(c) \tag{92}$$

该模型被用作许多论文的基准，这些论文是朴素贝叶斯分类器的单词级别，如下所示：

$$P(c_j \mid d_i; \hat{\theta}) = \frac{P(c_j \mid \hat{\theta})P(d_i \mid c_j; \hat{\theta}_j)}{P(d_i \mid \hat{\theta})} \tag{93}$$

4.4.2. 多项朴素贝叶斯分类器

如果文档的数量（n）符合k个类别，其中$k \in \{c_1, c_2, \ldots, c_k\}$，则作为输出的预测的类是$c \in C$。朴素贝叶斯算法可以描述如下：

$$P(c \mid d) = \frac{P(c)\prod_{w \in d} P(d \mid c)^{n_{wd}}}{P(d)} \tag{94}$$

其中，n_{wd}表示单词w出现在文档中的次数，P(w|c)是给定c类条件下观察单词w的概率。

P(w|c)计算如下所示：

$$P(w \mid c) = \frac{1 + \sum_{d \in D_c} n_{wd}}{k + \sum_{w'} \sum_{d \in D_c} n_{w'd}} \tag{95}$$

4.4.3. 用于不平衡类的朴素贝叶斯分类器

NBC的一个局限性是该技术在具有不平衡类的数据集上表现不佳。 Eibe Frank和Remco R. Bouckaert开发了一种方法，通过公式（96）在每个类中引入归一化，然后在NBC中使用质心分类器来表示不平衡类。类c的质心C_c在等式（97）中给出。

$$\alpha \times \frac{n_{wd}}{\sum_{w'} \sum_{d \in D_c} n_{w'd}} \tag{96}$$

$$c_c = \left\{ \frac{\sum_{d \in D_c} n_{w_1 d}}{\sqrt{\sum_w \left(\sum_{d \in D_c} n_{wd}\right)^2}}, \ldots, \frac{\sum_{d \in D_c} n_{w_i d}}{\sqrt{\sum_w \left(\sum_{d \in D_c} n_{wd}\right)^2}}, \ldots, \frac{\sum_{d \in D_c} n_{w_k d}}{\sqrt{\sum_w \left(\sum_{d \in D_c} n_{wd}\right)^2}} \right\} \tag{97}$$

评分函数定义为：

$$x_d.c_1 - x_d.c_2 \tag{98}$$

所以多项式朴素贝叶斯分类器的对数可以计算为：

$$\left[\log P(c_1)+\sum_{i=1}^{k} n_{w_i d}\log(P(w_i \mid c_1))\right]-\left[\log P(c_2)+\sum_{i=1}^{k} n_{w_i d}\log(P(w_i \mid c_2))\right] \tag{99}$$

使用公式(95)和(96)，如果$\alpha=1$ ，我们可以重新改写为：

$$P(w \mid c)=\frac{1+\frac{n_{wd}}{\sum_{w'}\sum_{d\in D_c} n_{w'd}}}{K+1} \tag{100}$$

关于：

$$\frac{\sum_{d\in D_c} n_{wd}}{\sum_{w'}\sum_{d\in D_c} n_{w'd}} << 1 \tag{101}$$

对于文本数据集和$\log(x+1)\approx x$和$x \ll 1$。在NBC的这种技术中，实验结果与质心分类非常相似。

4.4.4. 朴素贝叶斯算法的局限性

朴素贝叶斯算法也有一些局限性。NBC对数据分布的形状做出了强有力的假设。NBC也受到数据稀缺性的限制，对于特征空间中的任何可能值，一个似然值都必须由频域专家估计。

4.5. K-最近邻算法

k近邻算法(KNN)是一种非参数分类技术。该方法在过去几十年被应用于许多研究领域。

4.5.1. KNN基本概念

给定测试文档x，KNN算法在训练集中的所有文档中找到x的k个最近邻居，并根据k个近邻的类别对候选类别进行评分。x与每个邻居文档的相似性可以是邻居文档类别的得分。多个KNN文档可能属于同一类别； 在这种情况下，这些分数的总和将是类k相对于测试文档x的相似度得分。排序后，算法将候选者分配给测试文档x中得分最高的类。图11说明了KNN体系结构，但是为了简单起见，这个图是由2D数据集设计的(类似于文本数据集，并且具有更高的维度空间)。KNN的决策规则为:

$$\begin{aligned} f(x) &= arg\max_{j} S(x, C_j) \\ &= \sum_{d_i \in KNN} sim(x, d_i) y(d_i, C_j) \end{aligned} \tag{102}$$

其中S指的是关于$S(x, C_j)$的得分值，候选i到j的类的得分值，以及f(x)的输出是测试集文档的标签。

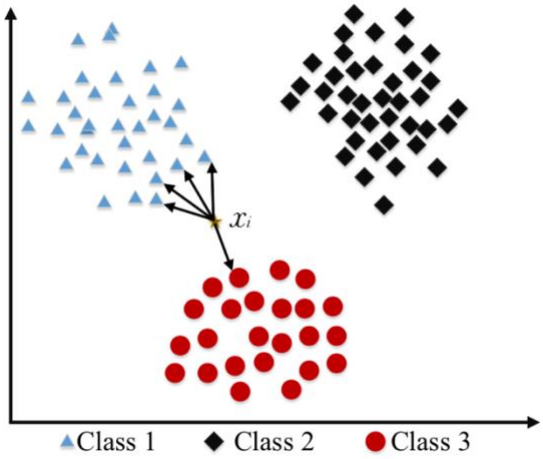

图11. 用于2D数据集和三个类的k-最近邻（KNN）模型的体系结构。

4.5.2. 权重调整的K-最近邻分类

权重调整的k-最近邻分类（WAKNN）是KNN的一种形式，其试图学习用于分类的权重向量。加权余弦度量计算如下：

$$\cos(x, y, w) \frac{\sum_{t \in T} (x_t \times w_t) \times (y_t \times w_t)}{\sqrt{\sum_{t \in T} (x_t \times w_t)^2} \times \sqrt{\sum_{t \in T} (y_t \times w_t)^2}} \tag{103}$$

其中T指的是单词集，x_t和y_t是TF，如第2节所述。对于训练模型（d∈D），设$N_d \in \{n_1, n_2, \ldots, n_k\}$是d的k近邻的集合。给定$N_d$，属于类c的d个邻居的相似度和定义如下：

$$S_c = \sum_{n_i \in N; C(n_i) = c} \cos(d, n_i, w) \tag{104}$$

总的相似度计算如下：

$$T = \sum_{c \in C} S_c \tag{105}$$

d的贡献按c类和T的S_c定义如下：

$$cont(d) = \begin{cases} 1 & \text{if } \forall c \in C, c \neq class(d), \\ & S_{class(d)} > S_s \text{ and } \frac{S_{class(d)}}{T} \leq p \\ 0 & \text{otherwise} \end{cases} \tag{106}$$

其中，cont(d)代表contribution(d)。

4.5.3. K-近邻算法的局限性

KNN是一种易于实现的分类方法，适用于任何类型的特征空间。该模型也自

然地处理多类案例。但是，KNN受限于大型搜索问题的数据存储限制以找到最近邻居。另外，KNN的性能取决于找到有意义的距离函数，因此使该技术成为非常依赖于数据的算法。

4.6. 支持向量机（SVM）

SVM的原始版本由Vapnik和Chervonenkis于1963年形成。Boser等人在20世纪90年代早期将这个版本改编成非线性公式。SVM最初是为二进制分类任务而设计的。然而，许多研究人员使用这种主导技术研究多类问题。图12显示了用于二维数据集的线性和非线性分类器。

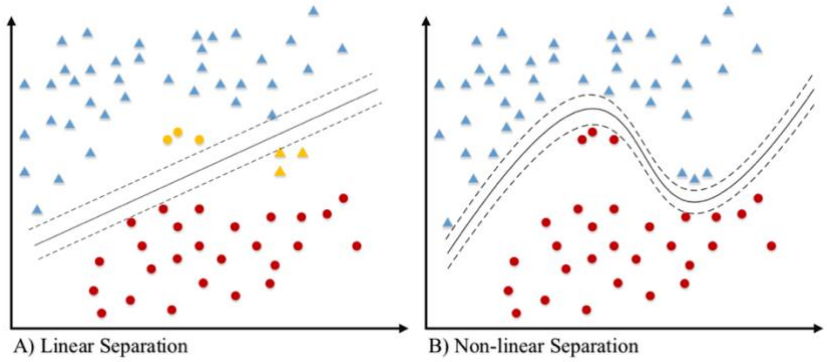

图12. 该图显示了2D数据集的线性和非线性支持向量机(SVM)（对于文本数据，我们有数千个维度）。 红色为第一类，蓝色为第二类，黄色为误分类数据点。

4.6.1. 二分类SVM

在文本分类的上下文中，令$x_1, x_2, \ldots, x_l$是属于类X的训练样本，其中X是R^N的紧凑子集。然后我们可以得到一个二元分类器:

$$\min \frac{1}{2}||w||^2 + \frac{1}{vl}\sum_{i=1}^{l}\xi_i - p \tag{107}$$

使服从：

$$(w.\Phi(x_i)) \geq p - \xi_i \;\; i = 1,2,\ldots,l \;\; \xi \geq 0 \tag{108}$$

如果w和p解决了这个问题，那么决策函数由下式给出：

$$f(x) = sign((w.\Phi(x)) - p) \tag{109}$$

4.6.2. 多分类SVM

由于SVM在传统用于二元分类，我们需要针对多类问题生成一个多SVM（MSVM）。One-vs-One是一种用于构建N(N-1)个分类器的多类SVM的技术，如下

所示：

$$f(x) = arg\max_i \left(\sum_j f_{ij}(x)\right) \tag{110}$$

解决k类问题的自然方法是同时构造所有k类的决策函数。通常，多类SVM是以下形式的优化问题：

$$\min_{w_1,w_2,..,w_k,\zeta} \frac{1}{2}\sum_k w_k^T w_k + C\sum_{(x_i,y_i)\in D}\zeta_i \tag{111}$$

$$st.\ w_{y_i}^T x - w_k^T x \le i - \zeta_i, \quad \forall(x_i,y_i)\in D, k\in\{1,2,...,K\}, k\ne y_i \tag{112}$$

其中（x_i，y_i）表示训练数据点，使得（x_i，y_i）∈D，C是惩罚参数，ζ是松弛参数，k代表该类。

另一种基于SVM的多类分类技术是All-vs-One。通过SVM进行特征提取通常使用以下两种方法之一：字序列特征提取和TF-IDF。但对于非结构化序列，如RNA和DNA序列，则使用字符串内核。然而，字符串内核可用于文档分类。

4.6.3. 字符串内核

文本分类也使用字符串内核进行了研究。字符串内核（SK）的基本思想是使用$\Phi(.)$来映射特征空间中的字符串。

谱核作为SK的一部分，已经应用于许多不同的应用中，包括文本、DNA和蛋白质分类。谱核的基本思想是将一个单词在字符串x_i中出现的次数作为一个特征映射来计算，其中定义了从$x \to R^{l^k}$的特征映射。

$$\Phi_k(x) = \Phi_j(x)_{j\in\Sigma^k} \tag{113}$$

$$where$$

$$\Phi_j(x) = number\ of\ j\ \ feature\ appears\ in\ x \tag{114}$$

特征映射$\Phi_i(x)$由序列x_i生成，并且核定义如下：

$$F = \Sigma^k \tag{115}$$

$$K_i(x,x') = <\Phi_i(x),\Phi_i(x')> \tag{116}$$

当应用于字符串序列分类时，SVM的主要限制是时间复杂度。使用字典大小Σ生成特征，F是特征的数量并且由等式（115）限定。内核计算与SP类似，并使用等式（116），最后使用等式（117）对内核进行归一化。

$$K^{Norm}(x,y) \leftarrow \frac{K(x,y)}{\sqrt{K(x,x)}\sqrt{K(y,y)}} \tag{117}$$

$$<f^x, f^y> = \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} h(u_i^{s_1}, u_j^{s_2}) \tag{118}$$

其中，两个序列u_i^{s1}和u_i^{s2}分别是s_1和s_2的长度。

4.6.4. 堆叠支持向量机(SVM)

堆叠SVM是一种分层分类方法，用于基于自顶向下的分级方法的分类树结构。该技术提供了单个SVM分类器的分层模型，因此通常比单SVM模型产生更准确的结果。如图13所示，叠加模型采用分层分类器，分层分类器包含多个层(在图中我们有两个层，比如mane域和子域)。

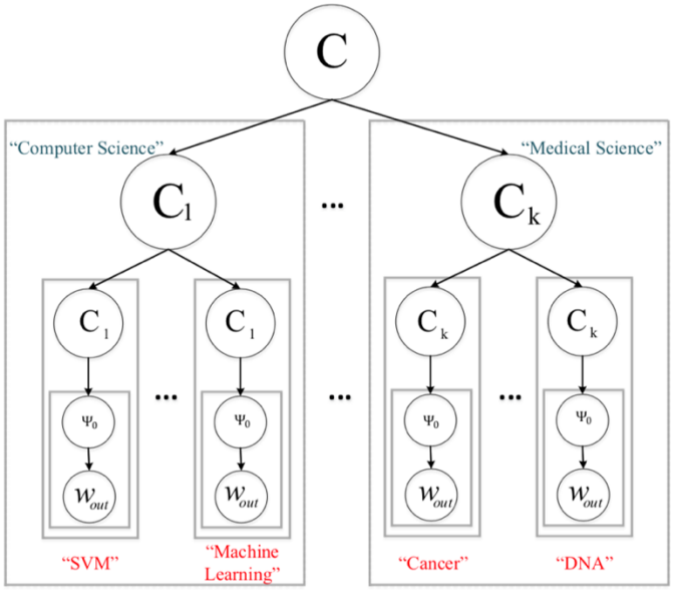

图13. 分层分类方法

4.6.5. 多实例学习(MIL)

多实例学习（MIL）是一种监督学习方法，通常被表示为为两种基于SVM的方法之一（mi-SVM和MI-SVM）。MIL将一组带标签的袋子作为输入而不是实例。如果袋中至少有一个带有阳性标签的实例，则袋标记为阳性;如果袋的所有实例均为阴性，则标记为阴性。然后，学习者试图推断出一个概念，该概念能够正确地为单个实例贴上标签。在统计模式识别中，假设标记模式的训练集是可用的，其中每对$(x_i, y_i) \in R^d \times Y$是独立地从未知分布生成的。目标是找到一个从模式到标签的分类器，即$f: R^d \to Y$。在MIL中，算法假定输入已知，其作为被分组成袋子$B_1, \ldots, B_m$的一组输入模式$x_1, \ldots, x_n$，其中，对于给定的索引$I \subseteq \{1, \ldots, n\}$，$B_I = \{x_i : i \in I\}$。每个袋子$B_I$与标签$Y_I$相关联，其中对于所有 i∈I，如果$y_i$ =-1则Y_I =-1，如果存在至少一个具有正标签的实例$x_i \in B_I$，则Y_I = 1。实例标签y_i和包标签Y_I之间的关系可以表示为$Y_I = max_{i \in I} y_i$或一组线性约束：

$$\begin{aligned} &\sum_{i \in I} \frac{y_i + 1}{2} \geq 1, \\ &\forall I \; s.t. \; Y_I = 1, \\ &y_i = -1, \forall I \; s.t. \; Y_I = -1. \end{aligned} \tag{119}$$

如果所有包B_I保持$sgn\ max_{\mathrm{i}\in\mathrm{I}}\mathrm{f}\ (x_i)\ =\ Y_I$，则判别函数f：X→R被称为关于多实例数据集的MI分离。

4.6.6. 支持向量机的局限性

自20世纪90年代推出以来，SVM一直是最有效的机器学习算法之一。然而，用于文本分类的SVM算法受到由维度高导致的结果不透明的限制。因此，它既不能以基于财务比率的参数函数来表示公司得分，也不能以任何其他函数形式来表示。另一个限制是可变的财务比率。

4.7. 决策树

用于文本和数据挖掘的一种早期分类算法是决策树。决策树分类器（DTC）在许多不同领域成功用于分类。该技术的结构是数据空间的分层分解。决策树作为分类任务由D.Morgan提出并由J.R.Quinlan发展。主要思想是基于分类数据点的属性创建树，但决策树的主要挑战是哪个属性或特征可以在父级中，哪个属于子级。为了解决这个问题，De Mántaras为树中的特征选择引入了统计建模。对于包含p个正例和n个反例的训练集：

$$H\left(\frac{p}{n+p},\frac{n}{n+p}\right)=-\frac{p}{n+p}\log_2\frac{p}{n+p}-\frac{n}{n+p}\log_2\frac{n}{n+p} \tag{120}$$

选择k值不同的属性A，将训练集E划分为子集{$\mathrm{E_1}$，$\mathrm{E_2}$，…，$\mathrm{E_k}$}。在尝试属性A（具有分支i=1,2,…,k）之后，期望熵（EH）保持不变：

$$EH(A)=\sum_{i=1}^{K}\frac{p_i+n_i}{p+n}H\left(\frac{p_i}{n_i+p_i},\frac{n_i}{n_i+p_i}\right) \tag{121}$$

此属性的信息增益（I）或熵减少为：

$$A(I)=H\left(\frac{p}{n+p},\frac{n}{n+p}\right)-EH(A) \tag{122}$$

选择具有最大信息增益的属性作为父节点。

决策树算法局限性

决策树是一种非常快速的学习和预测算法；但它对数据中的小扰动也非常敏感，并且很容易过拟合。这些影响可以通过验证方法和修剪来抵消，但这是一个灰色区域。该模型还存在样本外预测问题。

4.8.　随机森林

随机森林或随机决策森林技术是一种用于文本分类的集成学习方法。该方法使用t树并行，由T.Kam Ho于1995年引入。如图14所示，RF的主要思想是生成随机决策树。 这项技术于1999年由L.Breiman进一步发展，他发现RF作为边际度量(mg(X,Y))的收敛性如下:

$$mg(X,Y) = av_k I(h_k(X) = Y) - \max_{j \neq Y} av_k I(h_k(X) = j) \tag{123}$$

其中I(.)为指标函数。

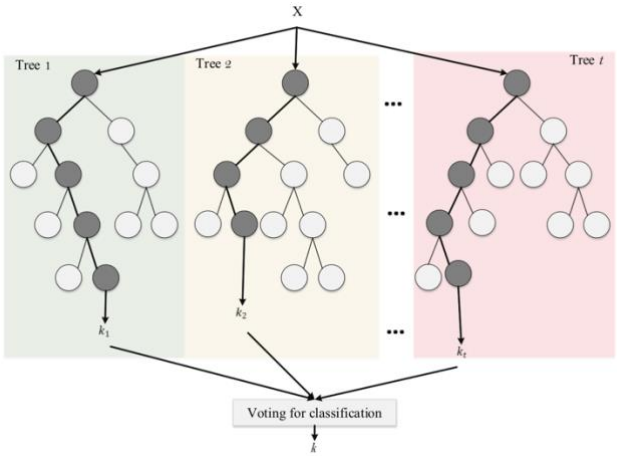

图14. 随机森林

4.8.1. 投票

将所有树木训练成森林后，根据投票结果预测如下：

$$\delta_V = arg\max_i \sum_{j:j \neq j} I_{\{r_{ij} > r_{ji}\}} \tag{124}$$

使得：

$$r_{ij} + r_{ji} = 1 \tag{125}$$

4.8.2. 决策森林局限性

随机森林（即决策树的集合）与其他技术（如深度学习）相比，训练文本数据集的速度非常快，但是一旦训练后创建预测的速度很慢。因此，为了实现更快的结构，必须减少森林中树木的数量，因为森林中更多的树木增加了预测步骤中的时间复杂性。

4.9.　条件随机场（CRF）

CRF是一种无向图形模型，如图15所示。CRF本质上是一种结合分类和图形建模优势的方法，它结合了对多变量数据进行紧凑建模的能力，以及利用高维特

征空间进行预测的能力（由于高特征空间，该模型对于文本数据非常强大）。CRF表示给定观察序列X的标签序列Y的条件概率，即P(Y|X)。CRF可以通过建模标签序列的条件概率而不是联合概率P（X,Y）来将复杂特征结合到观察序列中而不违反独立性假设。团势（即全连接子图）用于计算P(X|Y)。对于图中每一组的势函数，变量构型的概率对应于一系列非负势函数的乘积。每一个势函数计算的值等于某一特定构型对应小团体中变量的概率。那就是:

$$P(V) = \frac{1}{Z} \prod_{c \in cliques(V)} \psi(c) \tag{126}$$

其中Z是归一化项。条件概率 P(Y|X) 可以表示为：

$$P(Y|X) = \frac{1}{Z} \prod_{t=1}^{T} \psi(t, y_{t-1}, y_t, X) \tag{127}$$

给定潜在函数 $(\psi(t, y_{t-1}, y_t, X) = \exp(w.f(t, y_{t-1}, y_t, X)))$ ，条件概率可以被重写为：

$$P(Y|X) = \prod_{t=1}^{T} \exp(w.f(t, y_{t-1}, y_t, X)) \tag{128}$$

其中w是与由f计算的特征向量相关联的权向量。

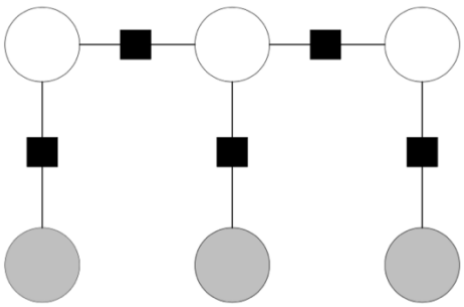

图15. 线性链条件随机场(CRF)。黑框是过渡集团

条件随机场局限性(CRF)

在CRF方面，CRF最明显的缺点是训练步骤的计算复杂度高，尤其是对于文本数据集，由于其特征空间较大。此外，该算法不执行不可见的字（即，在训练数据样本中不存在的单词）。

4.10. 深度学习

深度学习模型已经在许多领域取得了最先进的成果，包括各种各样的NLP应用程序。文本和文档分类的深度学习包括三个并行深度学习的基本体系结构。我们在下面详细描述每个单独的模型。

4.10.1. 深度神经网络

深度神经网络（DNN）旨在通过层的多连接来学习，每个层仅接收来自先前的连接，并且仅提供到隐藏部分中的下一层的连接。图16描绘了标准DNN的结构。输入包括输入特征空间（如第2节中所讨论的）与DNN的第一个隐藏层的连接。输入层可以通过TF-IDF、单词嵌入或其他特征提取方法构造。输出层等于多类分类的类数，或者只等于一个二分类的类数。在多类DNN中，生成每个学习模型（每层中的节点数量和层数被完全随机分配）。DNN的实现是一种判别训练模型，采用标准的反向传播算法，以sigmoid（公式（129）、ReLU（公式（130）为激活函数。多分类的输出层为Softmax函数（如公式（131））。

$$f(x) = \frac{1}{1+e^{-x}} \in (0,1) \tag{129}$$

$$f(x) = \max(0,x) \tag{130}$$

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \forall j \in \{1,\ldots,K\} \tag{131}$$

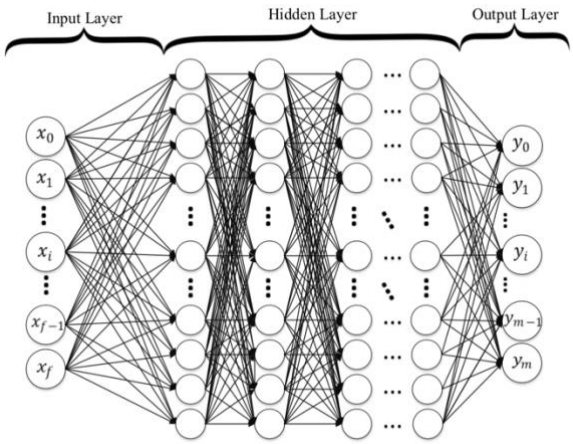

图16. 标准全连接深度神经网络（DNN）

给定一组示例对（x,y），x∈X，y∈Y，目标是使用隐藏层来学习这些输入和目标空间之间的关系。在文本分类应用程序中，输入是通过原始文本数据的矢量化生成的字符串。

4.10.2. 递归神经网络（RNN）

研究人员用于文本挖掘和分类的另一种神经网络架构是递归神经网络（RNN）。RNN为序列的先前数据点分配更多权重。因此，该技术是用于文本，字符串和顺序数据分类的强大方法。RNN以非常复杂的方法考虑先前节点的信

息，这允许对数据集的结构进行更好的语义分析。RNN主要通过使用LSTM或GRU进行文本分类，如图17所示，其中包含输入层（字嵌入），隐藏层和最终输出层。这种方法可以表述为：

$$x_t = F(x_{t-1}, u_t, \theta) \tag{132}$$

其中，x_t是时刻t的状态，u_t指的是第t步的输入。更具体地说，我们可以使用权重来表示公式(132)，参数化为：

$$x_t = \mathbf{W_{rec}}\sigma(x_{t-1}) + \mathbf{W_{in}}\mathbf{u_t} + \mathbf{b} \tag{133}$$

其中W_{rec}是指循环矩阵权重，W_{in}是指输入权重，b是偏差，σ表示逐元素函数。

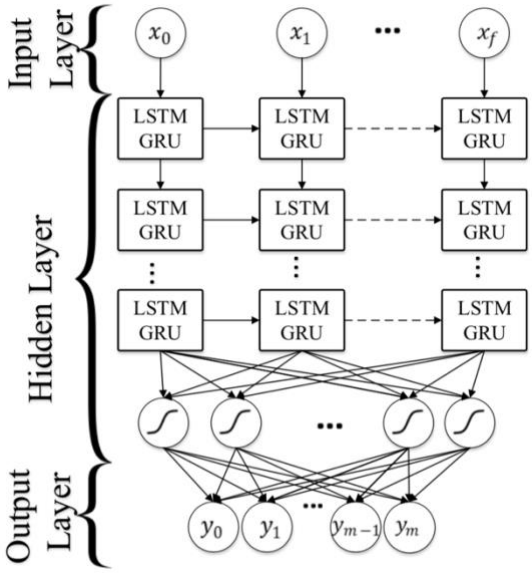

图17. 标准长短期记忆（LSTM）/ GRU递归神经网络。

图17说明了扩展的RNN架构。尽管有上述优点，但当梯度下降算法的误差通过网络传播时，RNN容易受到梯度消失和梯度爆炸的影响。

长短期记忆（LSTM）

LSTM由S.Hochreiter和J.Schmidhuber引入，并且后来被许多研究科学家所延伸。

LSTM是一种特殊类型的RNN，通过与基本RNN相比以更有效的方式保留长期依赖性来解决这些问题。LSTM在克服消失梯度问题方面特别有用。尽管LSTM具有类似于RNN的链状结构，但LSTM使用多个门来仔细调节允许进入每个节点状态的信息量。图18显示了LSTM模型的基本单元格。LSTM单元的逐步说明如

下：

$$i_t = \sigma(W_i[x_t, h_{t-1}] + b_i), \tag{134}$$
$$\tilde{C}_t = \tanh(W_c[x_t, h_{t-1}] + b_c), \tag{135}$$
$$f_t = \sigma(W_f[x_t, h_{t-1}] + b_f), \tag{136}$$
$$C_t = i_t * \tilde{C}_t + f_t C_{t-1}, \tag{137}$$
$$o_t = \sigma(W_o[x_t, h_{t-1}] + b_o), \tag{138}$$
$$h_t = o_t \tanh(C_t), \tag{139}$$

其中等式(134)表示输入门，等式(135)表示直接存储单元值，等式(136)定义遗忘门激活函数，等式(137）计算新存储单元值，等式(138)和(139)定义最终输出门值。在以上描述中，每个b表示偏置矢量，每个W表示权重矩阵，并且x_t表示在时间t对存储器单元的输入。此外，i，c，f，o指标分别指输入，单元记忆，遗忘和输出门。图18显示了这些门的结构的图形表示。

当后来的单词比以前的单词更具影响力时，RNN可能会有偏差。引入卷积神经网络（CNN）模型（第4.10.3节中讨论），通过部署最大池化层来确定文本数据中的区别短语来克服这种偏差。

门控循环单元（GRU）

GRU是由J.Chung等人和K.Cho等人制定的RNN的门控机制。GRU是LSTM架构的简化变体。但是，GRU与LSTM不同，因为它包含两个门，而GRU不具有内部存储器（即图18中的C_{t-1}）。此外，第二个非线性不应用（即，图18中的tanh）。GRU单元的逐步说明如下：

$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z), \tag{140}$$

其中z_t指t的更新门矢量，x_t代表输入矢量，W，U和b代表参数矩阵/矢量。 激活函数（σ_g）是sigmoid或ReLU，可以表示如下：

$$\tilde{r_t} = \sigma_g(W_r x_t + U_r h_{t-1} + b_r), \tag{141}$$

其中，r_t表示t的重置矢量，z_t是t的更新门矢量。

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \sigma_h(W_h x_t + U_h(r_t \circ h_{t-1}) + b_h) \tag{142}$$

其中，h_t是t的输出矢量，σ_h表示双曲正切函数。

4.10.3. 卷积神经网络(CNN)

卷积神经网络（CNN）是一种深度学习架构，通常用于分层文档分类。虽然最初是为图像处理而构建的，但CNN也已经有效地用于文本分类。在用于图像处理的基本CNN中，图像张量与一组大小为d×d的内核卷积。这些卷积层称为特征

图，可以堆叠起来，在输入端提供多个过滤器。为了降低计算复杂度，CNN使用池化来减小网络中从一层到下一层的输出大小。在保留重要特征的同时，使用不同的池化技术来减少输出。

最常见的池化方法是最大池化，其中选择池化窗口中的最大元素。为了将叠加的特征图合并输出提供给下一层，将图展平为一列。CNN中的最后一层通常是完全连接的。通常，在卷积神经网络的反向传播步骤期间，权重和特征检测滤波器都要进行调整。使用CNN进行文本分类时出现的潜在问题是“通道”的数量，Σ（特征空间的大小）。虽然图像分类应用通常具有很少的通道（例如，仅3个RGB通道），但是对于文本分类应用，Σ可能非常大（例如，50K），因此导致非常高的维度。图19展示了CNN的文本分类体系结构，其中包含了单词嵌入作为输入层，一维卷积层、一维池化层、全连接层和输出层。

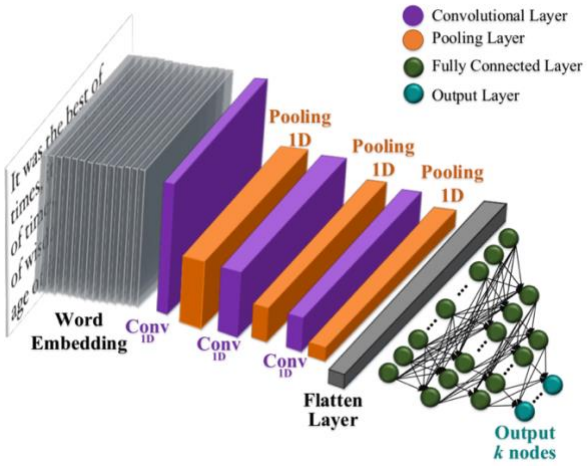

图19. 文本分类的卷积神经网络(CNN)

4.10.4. 深度置信网络(DBN)

深度置信网络（DBN）是一种深度学习结构，是由受限玻尔兹曼机（RBM）叠加组成。RBM是一种能够学习样本间概率分布的生成式人工神经网络。对比分歧（CD）算法是一种用于RBM的训练技术。

能量函数如下：

$$E(v,h) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_i \sum_j v_i w_{i,j} h_j \tag{143}$$

其中a_i是可见单元，b_i是指矩阵表示法中的隐藏单元。这个表达式可以简化为：

$$E(v,h) = -a^{\mathrm{T}} v - b^{\mathrm{T}} h - v^{\mathrm{T}} W h \tag{144}$$

给定隐藏单元h的配置定义如下：

$$P(v|h) = \prod_{i=1}^{m} P(v_i|h) \tag{145}$$

对于伯努利，可见单位的逻辑函数替换如下：

$$P(v_i^k = 1|h) = \frac{\exp(a_i^k + \Sigma_j W_{ij}^k h_j)}{\Sigma_{k'=1}^K \exp(a_i^{k'} + \Sigma_j W_{ij}^{k'} h_j)} \tag{146}$$

具有梯度下降的更新函数如下：

$$w_{ij}(t+1) = w_{ij}(t) + \eta \frac{\partial \log(p(v))}{\partial w_{ij}} \tag{147}$$

4.10.5. 分层注意网络(HAN)

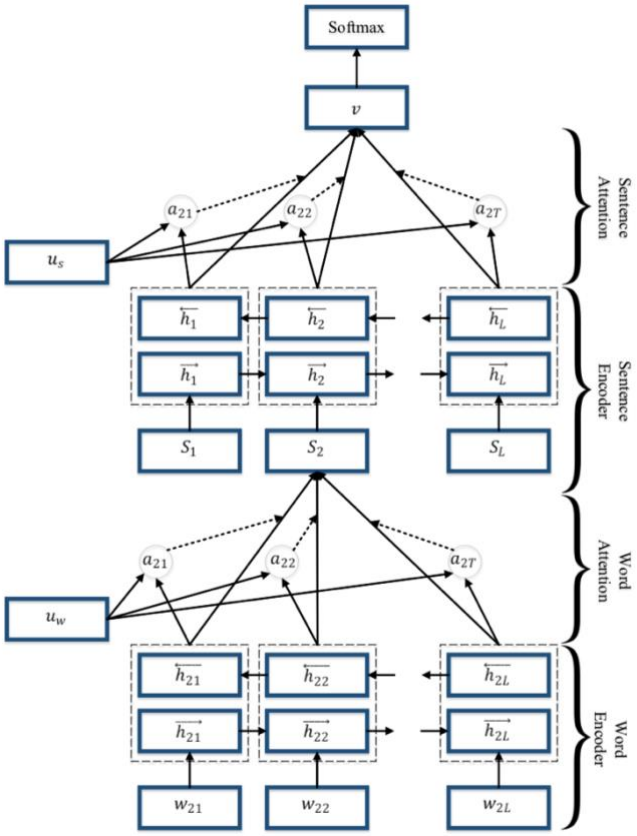

图20. 用于文档分类的分层注意网络。

层次注意网络(HAN)是一种成功的文本和文档分类的深层体系结构。该技术由Z.Yang等人和S.P.Hongsuck等人引进。HAN的结构侧重于文档级分类，其中文档具有L个句子并且每个句子包含T_i个单词，其中具有$t \int [1,T]$的w_{it}表示第i个句子中的单词。HAN架构如图20所示，其中较低级别包含单词编码和单词注意，而较高级别包含句子编码和句子注意。

4.10.6. 组合技术

许多研究人员将标准深度学习架构结合或连接起来，以便为分类任务开发具有更强大和精确架构的新技术。在本小节中，我们将介绍最近流行的深度学习架构和结构。

随机多模型深度学习（RMDL）

随机多模型深度学习(RMDL)是K.Kowsari等人提出的一种新的分类深度学习技术。RMDL可用于任何类型的数据集进行分类。该技术的概述如图21所示，它说明了使用multi-DNN，deep CNN和deep RNN的体系结构。所有这些深度学习多模型的层数和节点数是随机生成的(例如，RMDL中的9个随机模型由3个CNN，3个RNN和3个DNN构成，由于是随机生成的，因此它们都是唯一的)。

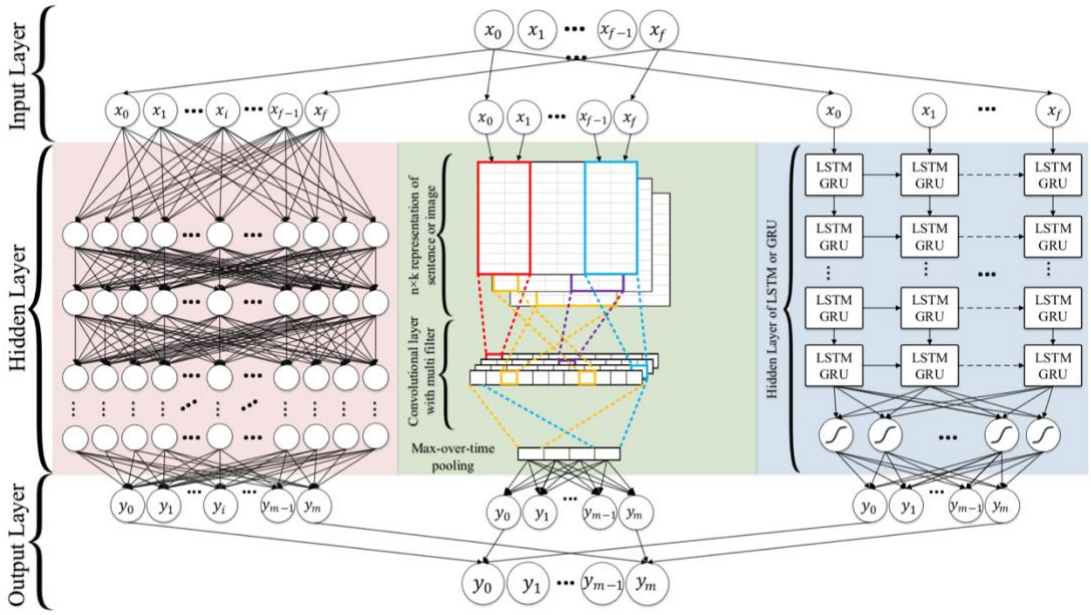

图21. 用于分类的随机多模型深度学习（RDML）架构。RMDL包括3个随机模型：深度神经网络（DNN）分类器（左），深度CNN分类器（中间）和深度递归神经网络（RNN）分类器（右）。每个单元可以是LSTM或GRU）。

$$M(y_{i1}, y_{i2}, \ldots, y_{in}) = \left\lfloor \frac{1}{2} + \frac{(\sum_{j=1}^{n} y_{ij}) - \frac{1}{2}}{n} \right\rfloor \tag{148}$$

其中，n是随机模型的数量，y_{ij}是第j个模型中第i个数据点的模型输出预测（等式（148）用于二进制分类，k∈{0或1}）。输出空间使用多数投票来计算y_i的最终值。因此，y_i给出如下：

$$\hat{y}_i = \left[\hat{y}_{i1} \ \ldots \ \hat{y}_{ij} \ \ldots \ \hat{y}_{in} \right]^T \tag{149}$$

其中n是随机模型的数量，y_{ij}表示对于模型j和y_i，标记$D_i \in \{x_i, y_i\}$的数据点（例如，文档）标签预测，j定义如下：

$$\hat{y}_{i,j} = arg\max_k [softmax(y^*_{i,j})] \tag{150}$$

在对所有RDL模型（RMDL）进行训练之后，使用对这些模型的输出进行多数投票来计算最终预测。使用具有不同优化器的多模型的主要思想是如果一个优化器不能很好地拟合特定数据集，具有n个随机模型的RMDL模型（其中一些可能使用不同的优化器）可以忽略效率不高的k个模型，当且仅当n>k时。使用多种技术的优化器（例如，SGD，Adam，RMSProp，Adagrad，Adamax）可以帮助RMDL模型更适用于任何类型的数据集。虽然在本研究中我们只使用了两个优化器(Adam和RMSProp)来评估模型，但是RMDL模型可以使用任何类型的优化器。在这一部分中，我们描述了在深度学习架构中使用的常见优化技术。

随机梯度下降(SGD)优化器

随机梯度下降(SGD)的基本方程如公式(151)所示。SGD对重新缩放的梯度使用动量，如公式(152)所示，用于更新参数。

$$\theta \leftarrow \theta - \alpha \nabla_\theta J(\theta, x_i, y_i) \tag{151}$$

$$\theta \leftarrow \theta - \left(\gamma\theta + \alpha \nabla_\theta J(\theta, x_i, y_i)\right) \tag{152}$$

RMSprop:

T.Tieleman和G.Hinton引入RMSprop作为一种新的优化器，它将权重的学习率除以该权重最近梯度大小的运行平均值。RMSprop动量法方程为:

$$v(t) = \alpha\, v(t-1) - \epsilon \frac{\partial E}{\partial w}(t) \tag{153}$$

$$\begin{aligned} \Delta w(t) &= v(t) \\ &= \alpha\, v(t-1) - \epsilon \frac{\partial E}{\partial w}(t) \\ &= \alpha\, \Delta v(t-1) - \epsilon \frac{\partial E}{\partial w}(t) \end{aligned} \tag{154}$$

RMSProp不做偏置校正，这在处理稀疏梯度时导致严重问题。

Adam 优化

Adam是另一个随机梯度优化器，它只使用梯度的前两个矩(v和m，如公式(155)-(158)所示)并计算它们的平均值。它能像RMSProp一样处理目标函数的非平稳，同时克服了RMSProp的稀疏梯度问题的局限性。

$$\theta \leftarrow \theta - \frac{\alpha}{\sqrt{\hat{v}} + \epsilon}\hat{m} \tag{155}$$

$$g_{i,t} = \nabla_\theta J(\theta_i, x_i, y_i) \tag{156}$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_{i,t} \tag{157}$$

$$m_t = \beta_2 v_{t-1} + (1-\beta_2) g_{i,t}^2 \tag{158}$$

其中m_t为第一个矩，v_t为第二个矩，均是被估计的量。$\widehat{\mathrm{m}_t} = \frac{\mathrm{m}_t}{1-\beta_1^t}$，$\widehat{\mathrm{v}_t} = \frac{\mathrm{v}_t}{1-\beta_2^t}$。

Adagrad:

Adagrad是一种新的子梯度方法家族，它动态地吸收数据几何知识，以实现更多的基于梯度的信息学习。

AdaGrad是SGD的扩展。在第k次迭代中，梯度定义为:

$$G^{(k)} = diag\left[\sum_{i=1}^{k} g^{(i)}(g^{(i)})^T\right]^{1/2} \tag{159}$$

对角矩阵：

$$G_{jj}^{(k)} = \sqrt{\sum_{i=1}^{k}(g_i^{(i)})^2} \tag{160}$$

更新规则：

$$\begin{aligned} x^{(k+1)} &= arg\min_{x\in X}\{\langle \nabla f(x^{(k)}), x\rangle + \\ &\qquad \frac{1}{2\alpha_k}||x - x^{(k)}||^2_{G^{(k)}}\} \\ &= x^{(k)} - \alpha B^{-1}\nabla f(x^{(k)}) \qquad (\text{if } X = R^n) \end{aligned} \tag{161}$$

Adadelta:

由M.D.Zeiler介绍的AdaDelta使用g_t指数衰减的平均值作为梯度的第二阶矩。此方法是Adagrad的更新版本，它仅依赖于一阶信息。Adadelta的更新规则是：

$$g_{t+1} = \gamma g_t + (1-\gamma)\nabla\mathcal{L}(\theta)^2 \tag{162}$$

$$x_{t+1} = \gamma x_t + (1-\gamma)v_{t+1}^2 \tag{163}$$

$$v_{t+1} = -\frac{\sqrt{x_t + \epsilon}\delta L(\theta_t)}{\sqrt{g_{t+1} + \epsilon}} \tag{164}$$

文本的分层深度学习（HDLTex）

文本分层深度学习（HDLTex）体系结构的主要贡献是文档的分层分类。传统的多类分类技术可以很好地适用于有限数量的类，但是随着类的数量增加，性能下降，这在分层组织的文档中是存在的。在这种分层深度学习模型中，通过创建专门针对其文档层次结构级别的深度学习方法的体系结构来解决该问题（例如，参见图22）。每种深度学习模型的HDLTex架构结构如下：

DNN: 8个隐层，每个隐层1024个cell。

RNN:在这个实现中使用了GRU和LSTM，100个cell的GRU带有两个隐藏层。

CNN:过滤器大小为{3, 4, 5, 6, 7}，最大池化大小为5，层大小为{128, 128, 128}，最大池化大小为{5, 5, 35}，CNN包含8个隐藏层。

使用以下参数构建所有模型：批大小= 128，学习参数= 0.001，β_1= 0.9，β_2= 0.999，ε= $1e^{08}$，衰减= 0.0，dropout= 0.5（DNN），dropout = 0.25（CNN和RNN）。

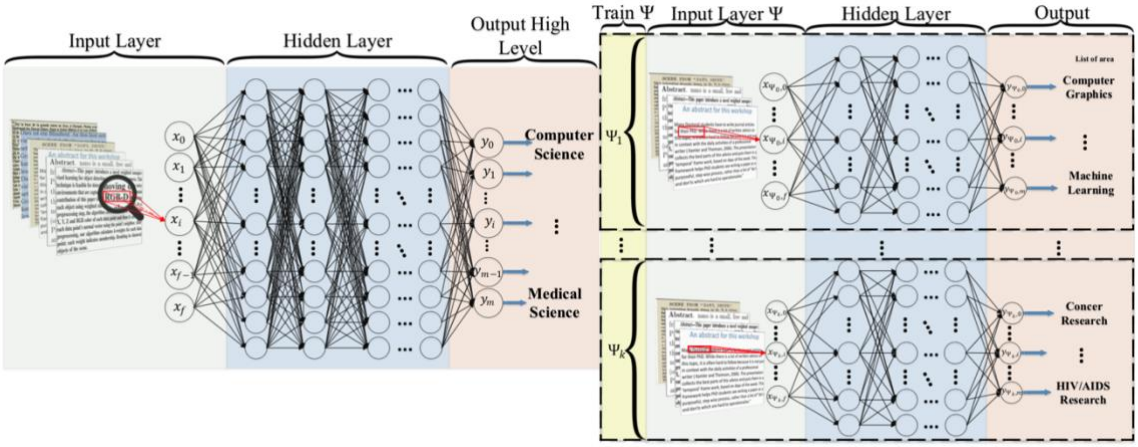

图22. HDLTex：用于文本分类的分层深度学习。DNN方法用于文本分类。上面的图描述了模型的父级，下面的图描述了作为父级输入文档的子级模型(ψ_i)。

HDLTex使用以下成本函数进行深度学习模型评估：

$$Acc(X) = \sum_{\varrho} \left[\frac{Acc(X_{\Psi_\varrho})}{k_\varrho - 1} \sum_{\Psi \in \{\Psi_1, .. \Psi_k\}} Acc(X_{\Psi_i}).n_{\Psi_k} \right] \quad (165)$$

其中，ρ是级别数，k表示每个级别的类数，Ψ是指层次模型中子级别的的类数。

其他技术

在本节中，我们将讨论结合深度学习架构的其他文本分类技术。循环卷积神经网络（RCNN）用于文本分类。RCNN可以使用循环结构捕获上下文信息，并使用CNN构建文本表示。该体系结构是RNN和CNN的组合，在模型中利用了这两种技术的优点。

C-LSTM是C. Zhou等人引入的另一种文本和文档分类技术，C-LSTM将CNN

和LSTM结合起来，利用卷积层学习短语级特征。这种体系结构将更高级别的表示序列输入到LSTM中，以学习长期依赖关系。

4.10.7. 深度学习局限性

深度学习（DL）的模型可解释性，特别是DNN，一直是建模时需要特征解释的用例的限制因素，并且许多医疗保健问题就是这种情况。这是由于科学家在工作中更倾向于使用传统的技术，如线性模型、贝叶斯模型、SVM、决策树等。神经网络中的权值是衡量每个神经元之间的连接强度，以找到重要的特征空间。如图23所示，模型越精确，可解释性越低，这意味着深度学习等复杂算法难以理解。

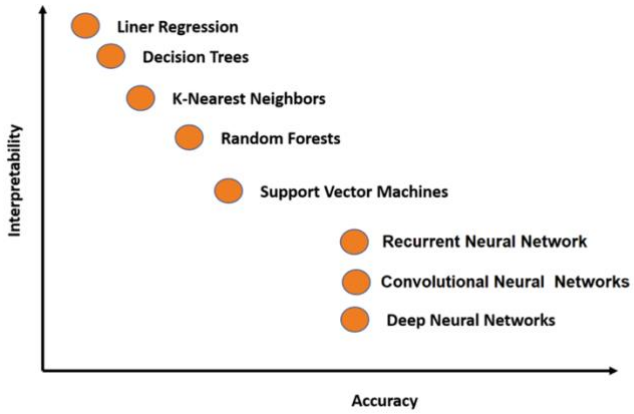

图23. 传统学习技术与深度学习技术的模型解释性比较。

深度学习（DL）是人工智能（AI）中最强大的技术之一，许多研究人员和科学家专注于深度学习架构，以提高该工具的鲁棒性和计算能力。然而，当应用于分类任务时，深度学习架构也具有一些缺点和局限性。该模型的主要问题之一是DL不能促进对学习的全面理论理解。DL方法的一个众所周知的缺点是它们的“黑箱”性质。也就是说，DL方法提出卷积输出的方法不容易理解。DL的另一个限制是它通常需要比传统机器学习算法更多的数据，这意味着该技术不能应用于小数据集上的分类任务。此外，DL分类算法所需的大量数据进一步加剧了计算训练步骤中的复杂性。

4.11. 文本分类的半监督学习

许多研究人员已经为标记和未标记的文档开发了许多有效的分类器。半监督学习是一种监督学习问题，它使用未标记的数据来训练模型。通常，当数据集的一小部分包含标记数据点且大量数据集不包含标签时，研究人员和科学家更喜欢使用半监督技术。大多数用于分类任务的半监督学习算法使用聚类 技术（通常

用于无监督学习）如下：最初，对于具有K＝K（类的数量）的D^T应用聚类技术，因为D^T已经标记了所有K类的样本。如果一个分区P_i已经标记了样本，那么，该群集上的所有数据点都属于该标签。

聚类技术的研究目标是确定我们是否在一个集群上标记了多个类，如果我们在一个集群中没有标记数据点会发生什么。在这一部分中，我们简要介绍了最受欢迎的半监督文本和文档分类技术。Chapelle和A.Zien通过低密度分离进行半监督分类，将图形距离计算与转导支持向量机（TSVM）训练相结合。Nigam等人利用期望最大化（EM）和文本分类领域中带标记和未标记数据的半监督学习的生成模型，开发了一种文本分类技术。Shi等人介绍了一种通过翻译特征在语言之间传递分类知识的方法。该技术使用EM算法，该算法自然地考虑了与单词翻译相关的模糊性。Su等人介绍了“半监督频率估计（SFE）”，一种用于大规模文本分类的MNBC方法。Zhou等人发明了一种新的深度学习方法，该方法使用模糊DBN进行半监督情感分类。该方法基于所学习的架构对每类评论采用模糊隶属函数。

5. 评估

在研究界，有一个共享和可比较的性能测量来评估算法是可取的。然而，实际上这些措施可能仅存在于少数方法中。评估文本分类方法时的主要问题是缺乏标准数据收集协议。即使存在共同的收集方法（例如，路透社新闻语料库），仅仅选择不同的训练和测试集也会引入模型性能的不一致性。方法评估的另一个挑战是能够比较单独实验中使用的不同性能指标。绩效评估通常评估分类任务绩效的具体方面，因此并不总是提供相同的信息。在本节中，我们将讨论评估指标和绩效指标，并重点介绍可以比较分类器性能的方法。由于不同评估指标的基础机制可能会有所不同，因此了解这些指标的确切含义以及它们尝试传达的信息类型对于可比性至关重要。这些指标的一些示例包括召回，精确度，准确度，F度量，微观平均值和宏观平均值。这些度量基于“混淆矩阵”（如图24所示），其包括真阳性（TP），假阳性（FP），假阴性（FN）和真阴性（TN）。这四个要素的重要性可能因分类应用而有所变化。所有预测中正确预测的分数称为准确度（公式（166））。正确预测已知阳性的部分称为灵敏度，即真阳性率或召回率（式（167））。正确预测阴性的比例称为特异性（式（168））。正确预测的阳性对所有阳性的比例称为精确度，即阳性预测值（等式（169））。

$$accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \tag{166}$$

$$sensitivity = \frac{TP}{(TP + FN)} \tag{167}$$

$$specificity = \frac{TN}{(TN + FP)} \tag{168}$$

$$precision = \frac{\sum_{l=1}^{L} TP_l}{\sum_{l=1}^{L} TP_l + FP_l} \tag{169}$$

$$recall = \frac{\sum_{l=1}^{L} TP_l}{\sum_{l=1}^{L} TP_l + FN_l} \tag{170}$$

$$F1 - Score = \frac{\sum_{l=1}^{L} 2TP_l}{\sum_{l=1}^{L} 2TP_l + FP_l + FN_l} \tag{171}$$

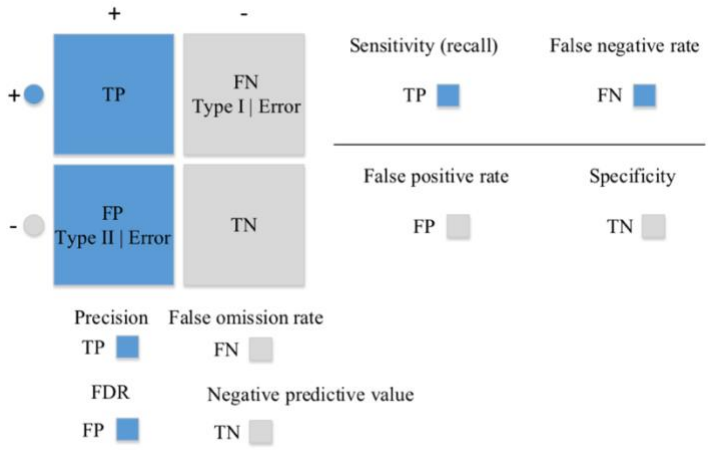

图24. 混淆矩阵

5.1. 宏观平均和微观平均

当多个两类分类器用于处理一个集合时，需要一个聚合度量。宏观平均给出了类的简单平均，而微观平均结合了类之间的每个文档决策，然后在合并的列联表格上输出一个有效度量。宏观平均结果可以计算如下：

$$B_{macro} = \frac{1}{q} \sum_{\lambda=1}^{q} B(TP_\lambda + FP_\lambda + TN_\lambda + FN_\lambda) \tag{172}$$

其中B是一个二元评估度量，是基于真阳性(TP)，假阳性(FP)，假阴性(FN)和真阴性(TN)计算的，并且$L = \{\lambda_j : j = 1 \ldots q\}$是所有标签的集合。

微观平均结果可以计算如下：

$$B_{macro} = B\left(\sum_{\lambda=1}^{q} TP_\lambda, \sum_{\lambda=1}^{q} FP_\lambda, \sum_{\lambda=1}^{q} TN_\lambda, \sum_{\lambda=1}^{q} FN_\lambda \right) \tag{173}$$

因此，微观平均分数为每个文档分配相同的权重，并且它被认为是每文档的平均值。另一方面，宏观平均分数为每个类别分配相同的权重而不考虑频率，因此，它是每个类别的平均值。

5.2. F_β分数

F_β是分类器评估最受欢迎的聚合评估指标之一。参数β用于平衡召回和精确度，定义如下：

$$F_\beta = \frac{(1+\beta^2)(precision \times recall)}{\beta^2 \times precision + recall} \tag{174}$$

对于常用的β=1，即F1，召回和精度给予相等的权重，等式(174)可以简化为：

$$F_1 = \frac{2TP}{2TP + FP + FN} \tag{175}$$

由于F_β基于召回率和精确度，因此它不能完全代表混淆矩阵。

5.3. 马修斯相关系数(MCC)

Matthews相关系数(MCC)在一个混淆矩阵中捕获所有数据，并度量二进制分类方法的质量。MCC可用于类规模不均匀的问题，仍被视为一种平衡措施。MCC的范围从-1到0（即，分别为分类总是错误的和总是正确的）。MCC的计算方法如下：

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{\begin{matrix}(TP+FP)\times(TP+FN)\times \\ (TN+FP)\times(TN+FN)\end{matrix}}} \tag{176}$$

在比较两个分类器时，一个可能使用MCC得分较高，另一个使用F1得分较高，因此一个特定指标无法捕获分类器的所有优点和缺点。

5.4. 接收者操作特征曲线

接收器操作特性（ROC）曲线是用于评估分类器的有价值的图形工具。然而，类不平衡（即，先验类概率的差异）可能导致ROC曲线不能很好地表示分类器性能。ROC曲线绘制真阳性率（TPR）和假阳性率（FPR）：

$$TPR = \frac{TP}{TP + FN} \tag{177}$$

$$FPR = \frac{FP}{FP + TN} \tag{178}$$

5.5. ROC曲线下面积（AUC）

ROC曲线下面积(AUC)表示ROC曲线下的整个面积。AUC利用了一些有用的特性，如方差分析(ANOVA)测试中灵敏度的提高、决策阈值的独立性、对先验类概率的不变性、以及对决策指数的正类和负类的指示性。

对于二分类任务，AUC可以表述为：

$$\begin{aligned} AUC &= \int_{-\infty}^{\infty} TPR(T)FPR'(T)dT \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(T' > T)f_1(T')f_0(T)dTdT' \\ &= P(X_1 > X_0) \end{aligned} \tag{179}$$

对于多级AUC，平均AUC可以定义如下：

$$AUC = \frac{2}{|C|(|C|-1)} \sum_{i=1}^{|C|} AUC_i \tag{180}$$

其中，C是类的数目。

Yang评估了文本分类的统计方法，并报告了在比较分类器算法时应考虑的以下重要因素：

- 通过方法和实验的比较评估，了解潜在性能变化的因素，并将导致未来更好的评估方法;
- 收集可变性的影响，例如在培训或测试集中包括未标记的文件，并将其视为负面实例，这可能是一个严重的问题;
- 类别排名评估和二元分类评估表明了分类器的有效性;
- 类别排名评估和二进制分类评估显示了分类器在交互式应用中的有用性，并分别强调它们在批处理模式下的使用。使用两种类型的性能测量来对分类器进行排序有助于检测阈值策略的影响;
- 评估大类别空间中分类器的可扩展性是一个很少被研究的领域。

6. 讨论

在本文中，我们旨在简要概述文本分类技术，并讨论相应的预处理步骤和评估方法。在本节中，我们将比较和对比这些技术和算法。此外，我们讨论了现有分类技术和评估方法的局限性。选择有效分类系统的主要挑战是理解不同管道步骤中可用技术的相似性和差异。

6.1. 文本和文档特征提取

表1 特征提取对比

模型	优点	局限性
加权词	● 易于计算 ● 使用它轻松计算2个文档之间的相似性 ● 提取文档中最具描述性的术语的基本度量 ● 使用未知单词（例如，语言中的新单词）	● 它不捕获文本（句法）中的位置 ● 它不捕获文本（语义）中的含义 ● 常用词对结果有影响（例如，“am”，“is”等）
TF-IDF	● 易于计算 ● 使用它轻松计算2个文档之间的相似性 ● 用于提取文档中描述性最强的术语的基本指标 ● 由于IDF，常用词不会而影响结果（例如，“am”，“is”等）	● 它不捕获文本（句法）中的位置 ● 它不捕获文本（语义）中的含义
Word2Vec	● 它捕获文本（句法）中的单词位置 ● 它捕获了单词（语义）中的含义	● 它无法从文本中捕获单词的含义(无法捕获多义词) ● 它不能从语料库中捕获词汇表外的单词
GloVe（预训练）	● 它捕获文本（句法）中单词的位置 ● 它捕获了单词（语义）中的含义 ● 在大型语料库上训练	● 它无法从文本中捕获单词的含义(无法捕获多义词) ● 存储内存消耗 ● 它不能从语料库中捕获词汇表外的单词
GloVe（训练）	● 它非常简单，例如，强制使用单词向量来捕获向量空间中的子线性关系(性能优于Word2vec) ● 降低高频词对的权重，例如“am”，“is”等停止词，不会影响训练进度	● 存储内存消耗 ● 需要大型语料库来学习 ● 它不能从语料库中捕获词汇表外的单词 ● 它无法从文本中捕获单词的含义(无法捕获多义词)
FastText	● 用于罕见的单词（在他们的字符n-gram中很少见，但仍与其他单词共享） ● •解决字符级别的n-gram词汇单词	● 它无法从文本中捕获单词的含义(无法捕获多义词) ● •存储内存消耗 ● •与GloVe和Word2Vec相比，计算成本更高
语境化的单词表示	● 它从文本中捕捉到单词的含义（包含上下文，处理多义词）	● 存储内存消耗 ● 显著提高下游任务的性能。与其他方法相比，计算方法更昂贵 ● 需要为所有LSTM和前馈层嵌入另一个单词 ● 它无法从语料库中捕获词汇表外的单词 ● 仅用于句子和文档级别(不适用于单个单词级别）

我们概述了以下两种主要特征提取方法：加权词（词袋）和词嵌入。单词嵌入技术通过考虑它们的出现和共现信息来学习单词序列。而且，这些方法是用于生成单词向量的无监督模型。相反，加权词特征基于对文档中的单词进行计数，并且可以用作单词表示的简单评分方案。每种技术都有其独特的局限性。加权单

词直接从单词计数空间计算文档相似度，这增加了大词汇表的计算时间。虽然独特单词的计数提供了相似性的独立证据，但它们不考虑单词之间的语义相似性（例如，“Hello”和“Hi”）。词嵌入方法解决了这个问题，但受限于需要大量文本数据集进行培训。因此，科学家们更喜欢使用预训练的字嵌入向量。但是，这种方法不适用于这些文本数据语料库中缺少的单词。例如，在一些短消息服务（SMS）数据集中，人们使用具有多种含义的单词，例如没有语义相似性的俚语或缩写。此外，缩写不包括在预训练的词嵌入向量中。为了解决这些问题，正如我们在第2节中讨论的，许多研究人员致力于文本清理。像GloVe，FastText和Word2Vec这样的单词嵌入技术都是基于该单词及其最近邻居进行训练的，这就包含了一个非常关键的限制(一个单词的含义在两个不同的句子中可能是不同的)。为了解决这个问题，科学家提出了一种称为语境化词语表示的新方法，它基于文档中单词的上下文进行训练。

如表1所示，我们比较和评估每种技术，包括加权词，TF-IDF，Word2Vec，Glove，FastText和语境化词表示。

6.2. 降维

在第3节中，我们概述了许多降维技术。在本节中，我们将讨论此步骤在文本分类系统的计算时间和鲁棒性方面的功效。降维主要用于改善计算时间和降低内存复杂度。

PCA试图找到包含可能的最高方差的数据集的正交投影，以便提取数据集的变量之间的线性相关性。PCA的主要局限性在于降维技术的计算复杂性。为了解决这个问题，科学家们引入了随机投影技术（第3节）。

LDA是一种有监督的降维技术，可以提高提取特征的预测性能。然而，LDA需要研究人员手动输入组件的数量，需要标记的数据，并生成不易解释的特性。

随机投影的计算速度比PCA快得多。但是，这种方法对小数据集表现不佳。

与其他DR方法相比，自动编码器需要更多的数据来训练，因此如果没有足够的数据，就不能作为通用的降维算法。

T-SNE主要用于文本和文档数据集中的数据可视化。

6.3. 现有的分类技术

在本节中，我们将讨论现有文本和文档分类算法的局限性和优势。 然后我们在两个表中比较最先进的技术。

6.3.1.局限性和优势

如表2和表3所示，Rocchio算法的局限性在于，用户只能使用该模型检索少量相关文档。此外，算法的结果说明了文本分类中的一些限制，可以通过考虑语义来解决。Boosting和bagging方法也有许多限制和缺点，例如计算复杂性和可解释性的损失。LR可以很好的预测分类输出。然而，该预测要求每个数据点是独立的，其试图基于一组独立变量来预测结果。 朴素贝叶斯算法也有一些局限性。NBC对数据分布的形状做出了强有力的假设。NBC也受到数据稀缺性的限制，其中特征空间中的任何可能值都必须由概率论估计。KNN是一种易于实现的分类方法，适用于任何类型的特征空间。该模型也可以自然地处理多类案例。KNN是一种易于实现的分类方法，适用于任何类型的特征空间。该模型也自然地处理多类案例。但是，KNN受限于大型搜索问题的数据存储约束以找到最近的邻居。此外，KNN的性能取决于找到一个有意义的距离函数，从而使该技术成为一种非常依赖数据的算法。SVM自20世纪90年代问世以来，一直是最高效的机器学习算法之一。然而，它们受到由于大量维度导致结果缺乏透明度的限制。因此，它既不能以基于财务比率的参数函数来表示公司得分，也不能以任何其他函数形式来表示。另一个限制是可变财务比率。决策树是一种学习和预测都非常快的算法，但是它对数据中的小扰动也非常敏感，容易过拟合。这些影响可以通过验证方法和剪枝来抵消，但这是一个灰色区域。该模型还存在样本外预测的问题。与其他技术相比，随机森林（即决策树的集合）训练速度非常快，但是一旦训练完成预测就很慢。因此，为了实现更快的结构，必须减少森林中树木的数量，因为森林中的树木越多，预测步骤的时间复杂度就越高。关于CRF，CRF最明显的缺点是训练步骤的计算复杂度高，并且该算法不能用未知单词执行（即，训练数据样本中不存在的单词）。深度学习（DL）是人工智能（AI）中最强大的技术之一，许多研究人员和科学家专注于深度学习架构，以提高该工具的鲁棒性和计算能力。然而，当应用于分类任务时，深度学习架构也具有一些缺点和局限性。 该模型的主要问题之一是DL不能促进对学习的全面理论理解。DL方法的一个众所周知的缺点是它们的“黑箱”性质。也就是说，DL方法提出卷积输出的方法不容易理解。DL的另一个限制是它通常比传统的机器学习算法需要更多的数据，这意味着这种技术不能应用于小数据集上的分类任务。另外，DL分类算法所需的大量数据进一步加剧了训练步骤期间的计算复杂性。

表2. 文本分类比较(Rocchio算法, boosting, bagging, 逻辑回归, 朴素贝叶斯分类器, k-最近邻, 支持向量机)。

模型	优点	缺点
Rocchio算法	● 易于实现 ● 计算上非常便宜 ● 相关性反馈机制(对不相关文档进行排序的好处)	● 用户只能检索少量相关文档 ● Rocchio经常将多模式类的类型错误分类 ● 这种技术不是很鲁棒 ● 此算法中的线性组合不适用于多类数据集
boosting和bagging	● 提高稳定性和准确性（利用集成学习的优势，在多个弱学习者中表现优于单个强学习者） ● 减小方差，有助于避免过拟合问题	● 计算复杂性 ● 可解释性丧失（如果模型数量很高，则理解模型非常困难） ● 需要仔细调整不同的超参数
逻辑回归	● 易于实现 ● 不需要太多的计算资源 ● 不需要对输入特性进行缩放(预处理) ● 不需要任何调优	● 它不能解决非线性问题 ● 预测要求每个数据点都是独立的 ● 试图基于一组独立变量预测结果
朴素贝叶斯分类器	● 它可以很好地处理文本数据 ● 易于实现 ● 与其他算法相比速度更快	● 对数据分布的形状有很强的假设 ● 受数据稀缺性的限制，特征空间中的任何可能值都必须由概率论估算
k-最近邻	● 对文本数据集有效 ● 非参数 ● 考虑文本或文档的更多本地特征 ● 自然地处理多类数据集	● 此模型的计算非常昂贵 ● 很难找到k的最优值 ● 寻找最近邻居的大型搜索问题的约束 ● 对于文本数据集，很难找到有意义的距离函数
支持向量机(SVM)	● SVM可以对非线性决策边界进行建模 ● 线性分离时，与逻辑回归类似 ● 对过度拟合问题（特别是对于高维空间的文本数据集）的鲁棒性	● 大量维度(尤其是文本数据)导致结果缺乏透明度。 ● 选择一个有效的内核函数是困难的(容易出现过度拟合/训练问题，这取决于内核) ● 存储复杂性

表3. 文本分类比较（决策树，条件随机场（CRF），随机森林和深度学习）。

模型	优点	缺点
决策树	● 能够轻松处理定性(分类)功能 ● 能够很好地将决策边界与特征轴相匹配 ● 决策树是一种非常快速的学习和预测算法	● 对角线决策边界的问题 ● 容易过拟合 ● 对数据中的微小扰动极其敏感 ● 样本外预测存在问题
条件随机场（CRF）	● 其功能设计灵活 ● 由于CRF计算全局最优输出节点的条件概率，因此它克服了标签偏差的缺点 ● 结合分类和图形建模的优点，结合了对多变量数据进行精确建模的能力	● 训练步骤计算复杂度高 ● 该算法对未知单词不执行 ● 在线学习的问题(当有新的数据可用时，很难重新训练模型)
随机森林	● 与其他技术相比，决策树的集合训练速度非常快 ● 减少方差（相对于常规树木） ● 不需要准备和预处理输入数据	● 一旦训练完成，创建预测的速度相当慢 ● 森林中树木越多，预测步骤的时间复杂度就越高 ● 不容易直观地解释 ● 容易发生过拟合 ● 需要选择森林中的树木数量
深度学习	● •灵活的功能设计（减少对功能工程的需求，这是机器学习实践中最耗时的部分之一） ● 能够适应新问题的架构 ● 能够处理复杂的输入输出映射 ● 能够轻松处理在线学习（当新数据可用时，重新训练模型很容易） ● 并行处理能力（它可以同时执行多个任务）	● 需要大量数据（如果您只有小样本文本数据，深度学习不可能超越其他方法）。 ● 训练计算成本极高。 ● 模型可解释性是深度学习中最重要的问题（深度学习大部分时间都是黑盒箱） ● 寻找有效的架构和结构仍然是这项技术的主要挑战

6.3.2. 最先进技术的比较

表4和表5比较了文本分类技术的标准:体系结构、作者、模型、新颖性、特征提取、细节、语料库、验证措施和每种技术的局限性。每个文本分类技术（系统）包含作为分类器算法的模型，并且还需要特征提取技术，其意味着将文本或文档数据集转换为数字数据（如第2节中所讨论的）。我们比较的另一个重要部分是用于评估系统的验证措施。

表 4. 文本分类技术的比较

模型	作者	结构	新颖性	特征提取	细节	语料库	验证方法	局限性
Rocchio算法	B.J. Sowmya et al.	分层Rocchio	对分层数据进行分类	TF-IDF	在 GPU 上使用CUDA来计算和比较距离。	维基百科	F1-Macro	仅适用于分层数据集并检索少量相关文档
Boosting	S. Bloehdorn et al.		AdaBoost具有语义功能	BOW	集成学习算法	路透社-21578	F1-Macro and F1-Micro	计算复杂性和可解释性的丧失
逻辑回归	A. Genkin et al.	贝叶斯逻辑回归	高维数据的逻辑回归分析	TF-IDF	它基于高斯先验和岭回归	RCV1-v2	F1-Macro	基于一组独立变量来预测输出
朴素贝叶斯	Kim, S.B et al.	权重增强方法	用于文本分类的多元泊松模型	权重词	每个文档的词频归一化来估计泊松参数	路透社-21578	F1-Macro	该方法对数据分布的形状做出了强有力的假设
SVM和KNN	K. Chen et al.	逆重力矩	引入TFGM（词频和反重力矩）	TF-IDF 和 TFIGM	合并一个统计模型来精确测量一个术语的类区分能力	20-Newsgroups 和路透社-21578	F1-Macro	未能捕捉多义词，也没有解决语义问题
支持向量机	H. Lodhi et al.	字符串的子序列内核	使用特殊内核	使用 TF-IDF 的相似性	内核是由长度为k的所有子序列生成的特征空间中的内积	路透社-21578	F1-Macro	结果缺乏透明度
条件随机场（CRF）	T. Chen et al.	BiLSTM-CRF	应用基于神经网络的序列模型，根据句子中出现的目标数量将固定的句子分为三类	单词嵌入	通过句型分类改善句子级别的情绪分析	客户审核	准确性	计算复杂度高，并且该算法对不可见字不执行

表5. 文本分类技术的比较（续）。

模型	作者	结构	新颖性	特征提取	细节	语料库	验证方法	局限性
深度学习	Z. Yang et al.	分层注意网络	具有分层结构	单词嵌入	在单词和句子级别应用两个级别的注意机制	Yelp，IMDB评论，亚马逊评论	准确性	仅适用于文档级别
深度学习	J. Chen et al	深度神经网络	卷积神经网络（CNN）使用2维TF-IDF特征	2维TF-IDF	语言攻击检测任务的一种新解决方案	推特评论	F1-Macro和F1-Micro	数据依赖于设计的模型架构
深度学习	M. Jiang et al.	深度置信网络	基于深度置信网络和softmax回归的混合文本分类模型。	DBN	DBN完成特征学习以解决高维和稀疏矩阵问题，并使用softmax回归对文本进行分类	路透社-21578和20-Newsgroup	误码率	计算很昂贵，并且模型的可解释性仍然是该模型的一个问题
深度学习	X. Zhang et al	CNN	用于文本分类的字符级卷积网络（ConvNets）	编码字符	字符级ConvNet包含6个卷积层和3个全连接层	Yelp，亚马逊评论和雅虎!答案数据集	相对误差	此模型仅用于发现其输入的位置不变特征
深度学习	K. Kowsari	集成深度学习算法（CNN，DNN和RNN）	解决了寻找最佳深度学习结构和架构的问题	TF-IDF和GloVe	随机多模型深度学习(RDML)	IMD评论，路透社-21578, 20-NewsGroup和WOS	准确性	计算很昂贵
深度学习	K. Kowsari	分层结构	采用一堆深度学习架构，以便在文档层次结构的每个级别提供专业的理解	TF-IDF和GloVe	用于文本分类的分层深度学习（HDLTex）	科学数据集	准确性	仅适用于分层数据集

6.4.　评估

文本分类器的实验评估度量了文本分类器的有效性（即，做出正确分类或分类决定的能力）。精确度和召回率被广泛用于衡量文本分类器的有效性。另一方面，准确度和误差（$\frac{FP+FN}{TP+TN+FP+FN} = 1 - $ 准确度）并未广泛用于文本分类应用，因为它们对由于分母的大值而导致的正确决策数量的变化不敏感。表 6 列出了上述每个指标的缺陷。

表 6. 指标缺陷

	局限性
准确度	没有给我们关于假阴性（FN）和假阳性（FP）的信息
灵敏度	不评估真阴性（TN）和 FP 以及任何将数据点预测为具有高灵敏度的阳性的分类器
特异性	与灵敏度类似，不考虑 FN 和 TP
精确度	不评估 TN 和 FN，被认为是非常保守的，并适用于最确定为阳性的情况

7. 文本分类用法

在 ML 和 AI 的早期历史中，文本分类技术主要用于信息检索系统。然而，随着时间的推移，技术的进步，文本分类和文档分类已被广泛应用于医学、社会科学、医疗保健、心理学、法律、工程等领域。在本节中，我们将重点介绍几个使用文本分类技术的领域。

7.1. 文本分类应用

7.1.1. 信息检索

信息检索是从大型文档集合中查找满足信息需求的非结构化数据的文档。随着在线信息的快速增长，特别是在文本格式中，文本分类已成为管理此类数据的重要技术。该领域中使用的一些重要方法有朴素贝叶斯，SVM，决策树，J48，KNN 和 IBK。文档和文本数据集处理最具挑战性的应用之一是应用文档分类方法进行信息检索。

7.1.2. 情感分析

情感分析是一种识别文本中观点、情感和主观性的计算方法。情感分类方法将与意见相关联的文档分类为正面或负面。假设是文档d表达对单个实体e的意见，并且意见是通过单个意见持有人h形成的。朴素贝叶斯分类和支持向量机是目前最流行的用于情绪分类的监督学习方法。在情感分类技术中，词汇及其出现频率、词性、观点词和短语、否定和句法依赖等特征都得到了应用。

7.1.4. 推荐系统

基于内容的推荐系统根据项目的描述和用户兴趣的资料向用户推荐项目。用户的个人资料可以从用户对项目的反馈（搜索查询或自我报告的历史）以及个人资料中的自解释功能（过滤或查询条件）中学习。这样，对这样的推荐系统的输入可以是半结构化的，使得一些属性从自由文本字段中提取而其他属性被直接指定。许多不同类型的文本分类方法，例如决策树，最近邻方法，Rocchio算法 ，线性分类器，概率方法和朴素贝叶斯，已被用于模拟用户的偏好。

7.1.5. 知识管理

文本数据库是重要的信息和知识来源。大部分企业信息(近80%)以文本数据格式(非结构化)存在。在知识摘要中，模式或知识是从可以是半结构化(例如，概念图表示)或结构化/关系(例如，数据表示)的直接形式推断出来的。给定的中间形式可以是基于文档的，使得每个实体代表特定域中感兴趣的对象或概念。文档分类是挖掘基于文档的中间形式的最常用方法之一。 在其他工作中，文本分类用于查找铁路事故原因与报告中相应描述之间的关系。

7.1.6. 文件摘要

用于文档概述的文本分类，其中文档的摘要可以使用未出现在原始文档中的单词或短语。由于在线信息的快速增长，多文档摘要也是必要的。因此，许多研究人员使用文本分类来关注此任务，以从文档中提取重要特征。

7.2. 文本分类支持

7.2.1. 健康

医学领域中的大多数文本信息以非结构化或叙述形式呈现，具有模糊的术语和印刷错误。在诊断和治疗的不同阶段，这种信息需要在病人-医生接触过程中立即获得。医学编码是医疗应用的一个领域，包括将医疗诊断分配给从大量类别中获得的特定类别值，其中文本分类技术非常有价值。在其他研究中，J.Zhang等人引入了Patient2Vec，来学习一种纵向电子健康记录（EHR）数据的可解释深层表示，该数据针对每位患者进行个性化处理。Patient2Vec是一种新的文本数据集特征嵌入技术，它可以学习基于递归神经网络和注意机制的EHR数据的个性化可解释的深层表示。文本分类在医学主题词(MeSH)和基因本体论(GO)的开发中也得到了应用。

7.2.2. 社会科学

在过去的几十年中，文本分类和文档分类越来越多地应用于理解人类行为。最近数据驱动的人类行为研究工作主要集中在挖掘非正式笔记和文本数据集中包含的语言中，包括短信服务（SMS），临床笔记，社交媒体等。这些研究主要集中在使用基于单词出现频率的方法（即单词出现在文档中的频率）或基于语言查询单词计数（LIWC）的特征，这是一个经过充分验证的具有心理相关性的单词类别词典。

7.2.3. 商业和营销

盈利的公司和组织正逐步将社交媒体用于营销目的。有利可图的公司和组织正逐步将社交媒体用于营销目的。从Facebook，Twitter等社交媒体开放采矿是公司快速增加利润的主要目标。文本和文档分类是公司更容易找到客户的有力工具。

7.2.4. 法律

政府机构已经产生了大量的法律文本信息和文件。检索此信息并自动对其进行分类不仅可以帮助律师，还可以帮助他们的客户。在美国，法律有五个来源：宪法，成文法，条约，行政法规和普通法。每年都会创建许多新的法律文件。 对这些文件进行分类是律师界面临的主要挑战。

8. 总结

分类任务是机器学习中最不可或缺的问题之一。随着文本和文档数据集的激增，监督机器学习算法的开发和记录成为一个迫切需要解决的问题，特别是对于文本分类。为这些信息提供更好的文档分类系统需要识别这些算法。但是，如果我们更好地理解特征提取方法以及如何正确评估它们，现有的文本分类算法的工作效率会更高。目前，文本分类算法可以主要按以下方式分类：（I）特征提取方法，例如词频-逆文档频率（TF-IDF），词频（TF），字嵌入（例如，Word2Vec，语境化单词表示，用词表示的全局向量（GloVe）和FastText），广泛用于学术和商业应用。在本文中，我们已经解决了这些技术。但是，文本和文档清理可以帮助应用程序的准确性和鲁棒性。我们描述了文本预处理步骤的基本方法。（II）降维方法，如主成分分析（PCA），线性判别分析（LDA），非负矩阵分解（NMF），随机投影，自动编码器和t分布随机邻域嵌入（t-SNE），可以用于减少现有文本分类算法的时间复杂度和存储器复杂度。在另一节中，介绍了最常见的降维方法。

（III）现有的分类算法是本文的重点，如Rocchio算法，bagging 和 boosting，逻辑回归（LR），朴素贝叶斯分类器（NBC），k近邻（KNN），支持向量机（SVM），决策树分类器（DTC），随机森林，条件随机场（CRF）和深度学习。（IV）评估方法，例如准确度，F_β，马修斯相关系数（MCC），接收者操作特性曲线（ROC）和曲线下面积（AUC），得到解释。使用这些指标，可以评估文本分类算法。（V）为了每种技术，解决了文本分类流水线的每个组件的严重限制（即，特征提取，降维，现有分类算法和评估）。最后，我们比较了本节中最常见的文本分类算法。（V）最后，文本分类作为一种应用和/或支持其他专业（例如法律，医学等）的用法，将在单独的部分中介绍。

在本次调查中，讨论了文本分类算法的最新技术和发展趋势。

参考文献

1. Jiang, M.; Liang, Y.; Feng, X.; Fan, X.; Pei, Z.; Xue, Y.; Guan, R. Text classification based on deep belief network and softmax regression. *Neural Comput. Appl.* **2018**, *29*, 61–70.

2. Kowsari, K.; Brown, D.E.; Heidarysafa, M.; Jafari Meimandi, K.; Gerber, M.S.; Barnes, L.E. HDLTex: Hierarchical Deep Learning for Text Classification. Machine Learning and Applications (ICMLA). In Proceedings of the 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), Cancun, Mexico, 18–21 December 2017.

3. McCallum, A.; Nigam, K. A comparison of event models for naive bayes text classification. In Proceedings of the AAAI-98 Workshop on Learning for Text Categorization, Madison, WI, USA, 26–27 July 1998; Volume 752, pp. 41–48.

4. Kowsari, K.; Heidarysafa, M.; Brown, D.E.; Jafari Meimandi, K.; Barnes, L.E. RMDL: Random Multimodel Deep Learning for Classification. In Proceedings of the 2018 International Conference on Information System and Data Mining, Lakeland, FL, USA, 9–11 April 2018; doi:10.1145/3206098.3206111.

5. Heidarysafa, M.; Kowsari, K.; Brown, D.E.; Jafari Meimandi, K.; Barnes, L.E. An Improvement of Data Classification Using Random Multimodel Deep Learning (RMDL). *IJMLC* **2018**, *8*, 298–310, doi:10.18178/ijmlc.2018.8.4.703.

6. Lai, S.; Xu, L.; Liu, K.; Zhao, J. Recurrent Convolutional Neural Networks for Text Classification. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; Volume 333, pp. 2267–2273.

7. Aggarwal, C.C.; Zhai, C. A survey of text classification algorithms. In *Mining Text Data*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 163–222.

8. Aggarwal, C.C.; Zhai, C.X. *Mining Text Data*; Springer: Berlin/Heidelberg, Germany, 2012.

9. Salton, G.; Buckley, C. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manag.* **1988**, *24*, 513–523.

10. Goldberg, Y.; Levy, O. Word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv* **2014**, arXiv:1402.3722.

11. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global Vectors for Word Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; Volume 14, pp. 1532–1543.

12. Mamitsuka, N.A.H. Query learning strategies using boosting and bagging. In *Machine Learning: Proceedings of the Fifteenth International Conference (ICML'98)*; Morgan Kaufmann Pub.: Burlington, MA, USA, 1998; Volume 1.

13. Kim, Y.H.; Hahn, S.Y.; Zhang, B.T. Text filtering by boosting naive Bayes classifiers. In

Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Athens, Greece, 24–28 July 2000; pp. 168–175.

14. Schapire, R.E.; Singer, Y. BoosTexter: A boosting-based system for text categorization. *Mach. Learn.* **2000**, *39*, 135–168.

15. Harrell, F.E. Ordinal logistic regression. In *Regression Modeling Strategies*; Springer: Berlin/Heidelberg, Germany, 2001; pp. 331–343.

16. Hosmer, D.W., Jr.; Lemeshow, S.; Sturdivant, R.X. *Applied Logistic Regression*; John Wiley & Sons: Hoboken, NJ, USA, 2013; Volume 398.

17. Dou, J.; Yamagishi, H.; Zhu, Z.; Yunus, A.P.; Chen, C.W. TXT-tool 1.081-6.1 A Comparative Study of the Binary Logistic Regression (BLR) and Artificial Neural Network (ANN) Models for GIS-Based Spatial Predicting Landslides at a Regional Scale. In *Landslide Dynamics: ISDR-ICL Landslide Interactive Teaching Tools*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 139–151.

18. Chen, W.; Xie, X.; Wang, J.; Pradhan, B.; Hong, H.; Bui, D.T.; Duan, Z.; Ma, J. A comparative study of logistic model tree, random forest, and classification and regression tree models for spatial prediction of landslide susceptibility. *Catena* **2017**, *151*, 147–160.

19. Larson, R.R. Introduction to information retrieval. *J. Am. Soc. Inf. Sci. Technol.* **2010**, *61*, 852–853.

20. Li, L.; Weinberg, C.R.; Darden, T.A.; Pedersen, L.G. Gene selection for sample classification based on gene expression data: Study of sensitivity to choice of parameters of the GA/KNN method. *Bioinformatics* **2001**, *17*, 1131–1142.

21. Manevitz, L.M.; Yousef, M. One-class SVMs for document classification. *J. Mach. Learn. Res.* **2001**, *2*, 139–154.

22. Han, E.H.S.; Karypis, G. Centroid-based document classification: Analysis and experimental results. In *European Conference on Principles of Data Mining and Knowledge Discovery*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 424–431.

23. Xu, B.; Guo, X.; Ye, Y.; Cheng, J. An Improved Random Forest Classifier for Text Categorization. *JCP* **2012**, *7*, 2913–2920.

24. Lafferty, J.; McCallum, A.; Pereira, F.C. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001), Williamstown, MA, USA, 28 June–1 July 2001.

25. Shen, D.; Sun, J.T.; Li, H.; Yang, Q.; Chen, Z. Document Summarization Using

Conditional Random Fields. *IJCAI* **2007**, *7*, 2862–2867.

26. Zhang, C. Automatic keyword extraction from documents using conditional random fields. *J. Comput. Inf. Syst.* **2008**, *4*, 1169–1180.

27. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444.

28. Huang, J.; Ling, C.X. Using AUC and accuracy in evaluating learning algorithms. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 299–310.

29. Lock, G. Acute mesenteric ischemia: classification, evaluation and therapy. *Acta Gastro-Enterol. Belg.* **2002**, *65*, 220–225.

30. Matthews, B.W. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim. Biophys. Acta (BBA)-Protein Struct.* **1975**, *405*, 442–451.

31. Hanley, J.A.; McNeil, B.J. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* **1982**, *143*, 29–36.

32. Pencina, M.J.; D'Agostino, R.B.; Vasan, R.S. Evaluating the added predictive ability of a new marker: From area under the ROC curve to reclassification and beyond. *Stat. Med.* **2008**, *27*, 157–172.

33. Jacobs, P.S. *Text-Based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval*; Psychology Press: Hove, UK, 2014.

34. Croft, W.B.; Metzler, D.; Strohman, T. *Search Engines: Information Retrieval in Practice*; Addison-Wesley Reading: Boston, MA, USA, 2010; Volume 283.

35. Yammahi, M.; Kowsari, K.; Shen, C.; Berkovich, S. An efficient technique for searching very large files with fuzzy criteria using the pigeonhole principle. In Proceedings of the 2014 Fifth International Conference on Computing for Geospatial Research and Application, Washington, DC, USA, 4–6 August 2014; pp. 82–86.

36. Chu, Z.; Gianvecchio, S.; Wang, H.; Jajodia, S. Who is tweeting on Twitter: Human, bot, or cyborg? In Proceedings of the 26th Annual Computer Security Applications Conference, Austin, TX, USA, 6–10 December 2010; pp. 21–30

37. Gordon, R.S., Jr. An operational classification of disease prevention. *Public Health Rep.* **1983**, *98*, 107.

38. Nobles, A.L.; Glenn, J.J.; Kowsari, K.; Teachman, B.A.; Barnes, L.E. Identification of Imminent Suicide Risk Among Young Adults using Text Messages. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, Montreal, QC, Canada, 21–26 April 2018; p. 413.

39. Gupta, G.; Malhotra, S. Text Document Tokenization for Word Frequency Count using Rapid Miner (Taking Resume as an Example). *Int. J. Comput. Appl.* **2015**, *975*, 8887.

40. Verma, T.; Renu, R.; Gaur, D. Tokenization and filtering process in RapidMiner. *Int. J. Appl. Inf. Syst.* **2014**, *7*, 16–18.

41. Aggarwal, C.C. *Machine Learning for Text*; Springer: Berlin/Heidelberg, Germany, 2018.

42. Saif, H.; Fernández, M.; He, Y.; Alani, H. On stopwords, filtering and data sparsity for sentiment analysis of twitter. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014), Reykjavik, Iceland, 26–31 May 2014.

43. Gupta, V.; Lehal, G.S. A survey of text mining techniques and applications. *J. Emerg. Technol. Web Intell.* **2009**, *1*, 60–76.

44. Dalal, M.K.; Zaveri, M.A. Automatic text classification: A technical review. *Int. J. Comput. Appl.* **2011**, *28*, 37–40.

45. Whitney, D.L.; Evans, B.W. Abbreviations for names of rock-forming minerals. *Am. Mineral.* **2010**, *95*, 185–187.

46. Helm, A. Recovery and reclamation: A pilgrimage in understanding who and what we are. In *Psychiatric and Mental Health Nursing: The Craft of Caring*; Routledge: London, UK, 2003; pp. 50–55.

47. Dhuliawala, S.; Kanojia, D.; Bhattacharyya, P. SlangNet: A WordNet like resource for English Slang. In Proceedings of the LREC, Portorož, Slovenia, 23–28 May 2016.

48. Pahwa, B.; Taruna, S.; Kasliwal, N. Sentiment Analysis-Strategy for Text Pre-Processing. *Int. J. Comput. Appl.* **2018**, *180*, 15–18.

49. Mawardi, V.C.; Susanto, N.; Naga, D.S. Spelling Correction for Text Documents in Bahasa Indonesia Using Finite State Automata and Levinshtein Distance Method. *EDP Sci.* **2018**, *164*, doi:10.1051/matecconf/201816401047.

50. Dziadek, J.; Henriksson, A.; Duneld, M. Improving Terminology Mapping in Clinical Text with Context-Sensitive Spelling Correction. In *Informatics for Health: Connected Citizen-Led Wellness and Population Health*; IOS Press: Amsterdam, The Netherlands, 2017; Volume 235, pp. 241–245.

51. Mawardi, V.C.; Rudy, R.; Naga, D.S. Fast and Accurate Spelling Correction Using Trie and Bigram. *TELKOMNIKA (Telecommun. Comput. Electron. Control)* **2018**, *16*, 827–833.

52. Spirovski, K.; Stevanoska, E.; Kulakov, A.; Popeska, Z.; Velinov, G. Comparison of different model's performances in task of document classification. In Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics, Novi Sad, Serbia, 25–27 June 2018; p. 10.

53. Singh, J.; Gupta, V. Text stemming: Approaches, applications, and challenges. *ACM*

Compu. Surv. (CSUR) **2016**, *49*, 45.

54. Sampson, G. *The'Language Instinct'Debate: Revised Edition*; A&C Black: London, UK, 2005.

55. Plisson, J.; Lavrac, N.; Mladenic´, D. A rule based approach to word lemmatization. In Proceedings of the 7th International Multi-Conference Information Society IS, **2004**,

56. Korenius, T.; Laurikkala, J.; Järvelin, K.; Juhola, M. Stemming and lemmatization in the clustering of finnish text documents. In Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management, Washington, DC, USA, 8–13 November 2004; pp. 625–633.

57. Caropreso, M.F.; Matwin, S. Beyond the bag of words: A text representation for sentence selection. In *Conference of the Canadian Society for Computational Studies of Intelligence*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 324–335.

58. Sidorov, G.; Velasquez, F.; Stamatatos, E.; Gelbukh, A.; Chanona-Hernández, L. Syntactic dependency-based n-grams as classification features. In *Mexican International Conference on Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 1–11.

59. Sparck Jones, K. A statistical interpretation of term specificity and its application in retrieval. *J. Doc.* **1972**, *28*, 11–21.

60. Tokunaga, T.; Makoto, I. Text categorization based on weighted inverse document frequency. In Proceedings of the Special Interest Groups and Information Process Society of Japan (SIG-IPSJ), **1994**.

61. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.

62. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 3111–3119.

63. Rong, X. word2vec parameter learning explained. *arXiv* **2014**, arXiv:1411.2738.

64. Maaten, L.V.D.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.

65. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching word vectors with subword information. *arXiv* **2016**, arXiv:1607.04606.

66. Melamud, O.; Goldberger, J.; Dagan, I. context2vec: Learning generic context embedding with bidirectional lstm. In Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, Berlin, Germany, 11–12 August 2016; pp.

51–61.

67. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations. *arXiv* **2018**, arXiv:1802.05365.

68. Abdi, H.; Williams, L.J. Principal component analysis. *Wiley Interdiscip. Rev. Comput. Stat.* **2010**, *2*, 433–459.

69. Jolliffe, I.T.; Cadima, J. Principal component analysis: A review and recent developments. *Philos. Trans. R. Soc. A* **2016**, *374*, 20150202.

70. Ng, A. Principal components analysis. Generative Algorithms, Regularization and Model Selection. *CS* **2015**, *229*, 71.

71. Cao, L.; Chua, K.S.; Chong, W.; Lee, H.; Gu, Q. A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine. *Neurocomputing* **2003**, *55*, 321–336.

72. Hérault, J. Réseaux de neurones à synapses modifiables: Décodage de messages sensoriels composites par une apprentissage non supervisé et permanent. *CR Acad. Sci. Paris* **1984**, *299* 525–528.

73. Jutten, C.; Herault, J. Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. *Signal Process.* **1991**, *24*, 1–10.

74. Hyvärinen, A.; Hoyer, P.O.; Inki, M. Topographic independent component analysis. *Neural Comput.* **2001**, *13*, 1527–1558.

75. Hyvärinen, A.; Oja, E. Independent component analysis: algorithms and applications. *Neural Netw.* **2000**, *13*, 411–430.

76. Sugiyama, M. Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. *J. Mach. Learn. Res.* **2007**, *8*, 1027–1061.

77. Balakrishnama, S.; Ganapathiraju, A. Linear discriminant analysis-a brief tutorial. *Inst. Signal Inf. Process.* **1998**, *18*, 1–8.

78. Sugiyama, M. Local fisher discriminant analysis for supervised dimensionality reduction. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006, pp. 905–912.

79. Pauca, V.P.; Shahnaz, F.; Berry, M.W.; Plemmons, R.J. Text mining using non-negative matrix factorizations. In Proceedings of the 2004 SIAM International Conference on Data Mining, Lake Buena Vista, FL, USA, 22–24 April 2004; pp. 452–456.

80. Tsuge, S.; Shishibori, M.; Kuroiwa, S.; Kita, K. Dimensionality reduction using non-negative matrix factorization for information retrieval. In Proceedings of the 2001 IEEE International Conference on Systems, Man, and Cybernetics, Tucson, AZ, USA, 7–10

October 2001; Volume 2, pp. 960–965.

81. Kullback, S.; Leibler, R.A. On information and sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86.

82. Johnson, D.; Sinanovic, S. Symmetrizing the kullback-leibler distance. *IEEE Trans. Inf. Theory* **2001**.

83. Bingham, E.; Mannila, H. Random projection in dimensionality reduction: Applications to image and text data. In Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 26–29 August 2001; pp. 245–250.

84. Chakrabarti, S.; Roy, S.; Soundalgekar, M.V. Fast and accurate text classification via multiple linear discriminant projections. *VLDB J.* **2003**, *12*, 170–185.

85. Rahimi, A.; Recht, B. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. *Adv. Neural Inf. Process. Syst.* **2009**, *21*, 1313–1320.

86. Morokoff, W.J.; Caflisch, R.E. Quasi-monte carlo integration. *J. Comput. Phys.* **1995**, *122*, 218–230.

87. Johnson, W.B.; Lindenstrauss, J.; Schechtman, G. Extensions of lipschitz maps into Banach spaces. *Isr. J. Math.* **1986**, *54*, 129–138, doi:10.1007/BF02764938.

88. Dasgupta, S.; Gupta, A. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Struct. Algorithms* **2003**, *22*, 60–65.

89. Mao, X.; Yuan, C. *Stochastic Differential Equations with Markovian Switching*; World Scientific: Singapore, 2016.

90. Goodfellow, I.; Bengio, Y.; Courville, A.; Bengio, Y. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016; Volume 1.

91. Wang, W.; Huang, Y.; Wang, Y.; Wang, L. Generalized autoencoder: A neural network framework for dimensionality reduction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA, 23–28 June 2014; pp. 490–497.

92. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. *Learning Internal Representations by Error Propagation*; Technical Report; California University San Diego, Institute for Cognitive Science: La Jolla, CA, USA, 1985.

93. Liang, H.; Sun, X.; Sun, Y.; Gao, Y. Text feature extraction based on deep learning: A review. *EURASIP J. Wirel. Commun. Netw.* **2017**, *2017*, 211.

94. Baldi, P. Autoencoders, unsupervised learning, and deep architectures. In Proceedings of ICML Workshop on Unsupervised and Transfer Learning, Bellevue, WA, USA, 2 July

2011; pp. 37–49.

95. AP, S.C.; Lauly, S.; Larochelle, H.; Khapra, M.; Ravindran, B.; Raykar, V.C.; Saha, A. An autoencoder approach to learning bilingual word representations. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 1853–1861.

96. Masci, J.; Meier, U.; Cires¸an, D.; Schmidhuber, J. Stacked convolutional auto-encoders for hierarchical feature extraction. In *International Conference on Artificial Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 52–59.

97. Chen, K.; Seuret, M.; Liwicki, M.; Hennebert, J.; Ingold, R. Page segmentation of historical document images with convolutional autoencoders. In Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR), Tunis, Tunisia, 23–26 August 2015; pp. 1011–1015.

98. Geng, J.; Fan, J.; Wang, H.; Ma, X.; Li, B.; Chen, F. High-resolution SAR image classification via deep convolutional autoencoders. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 2351–2355.

99. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 3104–3112.

100. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.

101. Hinton,G.E.;Roweis,S.T.Stochasticneighborembedding.*Adv.NeuralInf.Process.Syst.***2002**, *15*,857–864.

102. Joyce, J.M. Kullback-leibler divergence. In *International Encyclopedia of Statistical Science*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 720–722.

103. Rocchio, J.J. Relevance feedback in information retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*; Englewood Cliffs: Prentice-Hall, NJ, USA, 1971; pp. 313–323.

104. Partalas,I.;Kosmopoulos,A.;Baskiotis,N.;Artieres,T.;Paliouras,G.;Gaussier,E.;Androutsopoulos,I.; Amini, M.R.; Galinari, P. LSHTC: A benchmark for large-scale text classification. *arXiv* **2015**, arXiv:1503.08581.

105. Sowmya,B.;Srinivasa,K.Large scale multi-label text classificationof ahierarchical data set using Rocchio algorithm. In Proceedings of the 2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), Bangalore, India, 6–8 October 2016; pp. 291–296.

106. Korde,V.;Mahender,C.N.Text classification and classifiers: Asurvey. *Int. J. Artif. Intell. Appl.***2012**,*3*,85.

107. Selvi, S.T.; Karthikeyan, P.; Vincent, A.; Abinaya, V.; Neeraja, G.; Deepika, R. Text categorization using Rocchio algorithm and random forest algorithm. In Proceedings of the 2016 Eighth International Conference on Advanced Computing (ICoAC), Chennai, India, 19–21 January 2017; pp. 7–12.

108. Albitar,S.;Espinasse,B.;Fournier,S. Towards a Supervised Rocchio-based Semantic Classification of Web Pages. In Proceedings of the KES, San Sebastian, Spain, 10–12 September 2012; pp. 460–469.

109. Farzi,R.;Bolandi,V.Estimation of organic facies using ensemble methods in comparison with conventional intelligent approaches: A case study of the South Pars Gas Field, Persian Gulf, Iran. *Model. Earth Syst. Environ.* **2016**, *2*, 105.

110. Bauer,E.;Kohavi,R. An empiricalcomparison of voting classificational gorithms:Bagging,boosting, and variants. *Mach. Learn.* **1999**, *36*, 105–139.

111. Schapire,R.E. The strength of weak learnability.*Mach.Learn.***1990**,*5*,197–227.

112. Freund,Y. An improved boosting algorithm and its implications on learning complexity. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, Pittsburgh, PA, USA, 27–29 July 1992; pp. 391–398.

113. Bloehdorn,S.;Hotho,A. Boosting for text classification with semantic features. In *International Workshop on Knowledge Discovery on the Web*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 149–166.

114. Freund,Y.;Kearns,M.;Mansour,Y.;Ron,D.;Rubinfeld,R.;Schapire,R.E. Efficient algorithms for learning to play repeated games against computationally bounded adversaries. In Proceedings of the 36th Annual Symposium on Foundations of Computer Science, Milwaukee, WI, USA, 23–25 October 1995; pp. 332–341.

115. Breiman,L. Bagging predictors. *Mach.Learn.***1996**,*24*,123–140.

116. Geurts,P.Some enhancements of decision tree bagging. In *European Conferenceon Principles of Data Mining and Knowledge Discovery*; Springer: Berlin/Heidelberg, Germany, 2000; pp. 136–147.

117. Cox,D.R. *Analysis of Binary Data*; Routledge:London,UK,2018.

118. Fan,R.E. ;Chang,K.W.;Hsieh,C.J.;Wang,X.R.;Lin,C.J.LIBLINEAR: A library for large linear classification. *J. Mach. Learn. Res.* **2008**, *9*, 1871–1874.

119. Genkin, A.; Lewis, D.D.; Madigan, D. Large-scale Bayesian logistic regression for text categorization. *Technometrics* **2007**, *49*, 291–304.

120. Juan, A.; Vidal, E. On the use of Bernoulli mixture models for text classification. *Pattern Recogn.* **2002**, *35*, 2705–2710.

121. Cheng, W.; Hüllermeier, E. Combining instance-based learning and logistic regression for multilabel classification. *Mach. Learn.* **2009**, *76*, 211–225.

122. Krishnapuram,B. ;Carin,L.;Figueiredo,M.A.;Hartemink,A.J.Sparsemultinomiallogistic regression:Fast algorithms and generalization bounds. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 957–968.

123. Huang, K. Unconstrained Smartphone Sensing and Empirical Study for Sleep Monitoring and Self-Management. Ph.D. Thesis, University of Massachusetts Lowell, Lowell, MA, USA, 2015.

124. Guerin,A. *Using Demo graphic Variables and In-College Attributes to Predict Course-Level Retention for Community College Spanish Students*; Northcentral University: Scottsdale, AZ, USA, 2016.

125. Kaufmann, S. CUBA: Artificial Conviviality and User-Behaviour Analysis in Web-Feeds. PhD Thesis, Universität Hamburg, Hamburg, Germany 1969.

126. Porter,M.F.Analgorithmforsuffixstripping.*Program***1980**,*14*,130–137.

127. Pearson,E.S.Bayes'theorem,examinedinthelightofexperimentalsampling.*Biometrika***1925**, *17*,388–442.

128. Hill, B.M. Posterior distribution of percentiles: Bayes' theorem for sampling from a population. *J. Am. Stat. Assoc.* **1968**, *63*, 677–691.

129. Qu,Z.;Song,X.;Zheng,S.;Wang,X.;Song,X.;Li,Z.ImprovedBayesMethodBasedonTF-IDFFeatureand Grade Factor Feature for Chinese Information Classification. In Proceedings of the 2018 IEEE International Conference on Big Data and Smart Computing (BigComp), Shanghai, China, 15–17 January 2018; pp. 677–680.

130. Kim,S.B.;Han,K.S.;Rim,H.C.;Myaeng,S.H.Someeffectivetechniquesfornaivebayestextcla ssification. *IEEE Trans. Knowl. Data Eng.* **2006**, *18*, 1457–1466.

131. Frank,E.;Bouckaert,R.R.Naivebayesfortextclassificationwithunbalancedclasses.In*Europe anConferenceon Principles of Data Mining and Knowledge Discovery*; Springer: Berlin/Heidelberg, Germany, 2006, pp. 503–510.

132. Liu,Y.;Loh,H.T.;Sun,A.Imbalancedtextclassification:Atermweightingapproach.*ExpertSys t.Appl.***2009**, *36*, 690–701.

133. Soheily-Khah,S. ; Marteau,P.F.;Béchet,N. Intrusion detection in network systems through hybrid supervised and unsupervised mining process-a detailed case study on the ISCX benchmark data set. *HAL* **2017**, doi:10.1016/j.jisa.nnnn.nn.nnn.

134. Wang, Y.; Khardon, R.; Protopapas, P. Nonparametric bayesian estimation of periodic light curves. *Astrophys. J.* **2012**, *756*, 67.

135. Ranjan,M.N.M.;Ghorpade,Y.R.;Kanthale,G.R.;Ghorpade,A.R.;Dubey,A.S.DocumentClassificationusing LSTM Neural Network. *J. Data Min. Manag.* **2017**, *2*.

136. Jiang,S.;Pang,G.;Wu,M.;Kuang,L.AnimprovedK-nearest-neighboralgorithmfortextcategorization. *Expert Syst. Appl.* **2012**, *39*, 1503–1509.

137. Han, E.H.S.; Karypis, G.; Kumar, V. Text categorization using weight adjusted k-nearest neighbor classification. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*; Springer: Berlin/Heidelberg, Germany, 2001; pp. 53–65.

138. Salton,G. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of*; Addison-Wesley:Reading, UK, 1989.

139. Sahgal,D.; Ramesh, A. On Road Vehicle Detection Using Gabor Wave let Features with Various Classification Techniques. *IJEETC* **2015**, *1*, 10.1109/ICDSP.2002.1028263 .

140. Patel,D.;Srivastava,T.Ant Colony Optimization Model for Discrete Tomography Problems. In *Proceedings of the Third International Conference on Soft Computing for Problem Solving*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 785–792.

141. Sahgal, D.; Parida, M. Object Recognition Using Gabor Wavelet Features with Various Classification Techniques. In *Proceedings of the Third International Conference on Soft Computing for Problem Solving*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 793–804.

142. Sanjay,G.P.;Nagori,V.;Sanjay,G.P.;Nagori,V. Comparing Existing Methods for Predicting the Detection of Possibilities of Blood Cancer by Analyzing Health Data. *Int. J. Innov. Res. Sci. Technol.* **2018**, *4*, 10–14.

143. Vapnik,V.;Chervonenkis,A.Y.A class of algorithms for pattern recognition learning.*Avtomat.Telemekh***1964**, *25*, 937–945.

144. Boser,B.E.;Guyon,I.M.;Vapnik,V.N.A training algorithm for optimal margin classifiers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, Pittsburgh, PA, USA, 27–29 July 1992; pp. 144–152.

145. Bo, G.; Xianwu,H.SVM Multi-Class Classification. *J. Data Acquis. Process.* **2006**, *3*,017.

146. Mohri,M.; Rostamizadeh,A.; Talwalkar,A. *Foundations of Machine Learning*; MIT Press: Cambridge, MA, USA, 2012.

147. Chen, K.; Zhang, Z.; Long, J.; Zhang, H. Turning from TF-IDF to TF-IGM for term weighting in text classification. *Expert Syst. Appl.* **2016**, *66*, 245–260.

148. Weston,J.;Watkins,C.*Multi-Class Support Vector Machines*; Technical Report CSD-TR-98-04; Departmentof Computer Science, Royal Holloway, University of London: Egham,

UK, 1998.

149. Zhang, W.; Yoshida, T.; Tang, X. Text classification based on multi-word with support vector machine. *Knowl.-Based Syst.* **2008**, *21*, 879–886.

150. Lodhi,H.;Saunders,C.;Shawe-Taylor,J.; Cristianini,N.; Watkins,C. Text classification using string kernels. *J. Mach. Learn. Res.* **2002**, *2*, 419–444.

151. Leslie, C.S.; Eskin, E.; Noble, W.S. The spectrum kernel: A string kernel for SVM protein classification. *Biocomputing* **2002**, *7*, 566–575.

152. Eskin, E.; Weston, J.; Noble, W.S.; Leslie, C.S. Mismatch string kernels for SVM protein classification. *Adv. Neural Inf. Process. Syst.* **2002**, *15*, 1417–1424.

153. Singh,R.;Kowsari,K.;Lanchantin,J.;Wang,B.;Qi,Y.GaKCo:AFast and Scalable Algorithm for Calculating Gapped k-mer string Kernel using Counting. *bioRxiv* **2017**, doi:10.1101/329425.

154. Sun,A.;Lim,E.P.Hierarchical text classification and evaluation. In Proceedings of the Proceedings IEEE International Conference on Data Mining (ICDM 2001), San Jose, CA, USA, 29 November–2 December 2001; pp. 521–528.

155. Sebastiani,F. Machine learning in automated text categorization. *ACM Compu. Surv.(CSUR)***2002**,*34*,1–47.

156. Maron,O.;Lozano-Pérez,T. A frame work for multiple-instance learning. *Adv.NeuralInf.Process.Syst.***1998**, *10*, 570–576.

157. Andrews, S.; Tsochantaridis, I.; Hofmann, T. Support vector machines for multiple-instance learning. *Adv. Neural Inf. Process. Syst.* **2002**, *15*, 577–584.

158. Karamizadeh, S.; Abdullah, S.M.; Halimi, M.; Shayan, J.; Javad Rajabi, M. Advantage and drawback of support vector machine functionality. In Proceedings of the 2014 International Conference on Computer, Communications, and Control Technology (I4CT), Langkawi, Malaysia, 2–4 September 2014; pp. 63–65.

159. Guo, G. Soft biometrics from face images using support vector machines. In *Support Vector Machines Applications*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 269–302.

160. Morgan,J.N.;Sonquist,J.A.Problems in the analysis of survey data,and aproposal. *J.Am.Stat.Assoc.***1963**, *58*, 415–434.

161. Safavian,S.R.;Landgrebe,D.A survey of decision tree classifier methodology. *IEEE Trans.Syst.ManCybern.* **1991**, *21*, 660–674.

162. Magerman,D.M. Statistical decision-tree models for parsing.In Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics, Cambridge, MA, USA, 26–30 June 1995; Association for Computational Linguistics: Stroudsburg, PA, USA, 1995;

pp. 276–283.

163. Quinlan,J.R.Induction of decision trees.*Mach.Learn.***1986**,*1*,81–106.

164. DeMántaras, R.L. A distance-based attributes election measure for decision tree induction. *Mach.Learn.* **1991**, *6*, 81–92.

165. Giovanelli, C.;Liu,X.;Sierla,S.;Vyatkin,V.; Ichise,R. Towards an aggregator that exploits big data to bid on frequency containment reserve market. In Proceedings of the 43rd Annual Conference of the IEEE Industrial Electronics Society (IECON 2017), Beijing, China, 29 October–1 November 2017; pp. 7514–7519.

166. Quinlan,J.R.Simplifying decision trees.*Int.J.Man-Mach.Stud.***1987**,*27*,221–234.

167. Jasim,D.S. Data Mining Approach and Its Application to Dresses Sales Recommendation. **2018**.

168. Ho, T.K. Random decision forests. In Proceedings of 3rd International Conference on Document Analysis and Recognition, Montreal, QC, Canada 14–16 August 1995; Volume 1, pp. 278–282, doi:10. 1109/ICDAR. 1995. 598994.

169. Breiman,L.*RandomForests*; UC Berkeley TR567; University of California: Berkeley, CA,USA,1999.

170. Wu, T.F.; Lin, C.J.; Weng, R.C. Probability estimates for multi-class classification by pairwise coupling. *J. Mach. Learn. Res.* **2004**, *5*, 975–1005.

171. Bansal, H.; Shrivastava, G.; Nhu, N.; Stanciu, L. *Social Network Analytics for Contemporary Business Organizations*; IGI Global: Hershey, PA, USA, 2018.

172. Sutton,C.;McCallum,A. An introduction to conditional randomf ields. *Found. Trends ®Mach.Learn.***2012**, *4*, 267–373.

173. Vail,D.L.;Veloso,M.M.;Lafferty,J.D. Conditional random fields for activity recognition. In Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems, Honolulu, HI, USA, 14–18 May 2007; p. 235.

174. Chen, T.; Xu, R.; He, Y.; Wang, X. Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN. *Expert Syst. Appl.* **2017**, *72*, 221–230.

175. Sutton,C.; McCallum, A. An introduction to conditional random fields for relational learning. In *Introduction to Statistical Relational Learning*; MIT Press: Cambridge, MA, USA, 2006; Volume 2.

176. Tseng,H.;Chang,P.;Andrew,G.;Jurafsky,D.;Manning,C. A conditional random field word segmenter for sigh an bake off 2005. In Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing, Jeju Island, Korea, 14–15 October 2005.

177. Nair,V.;Hinton,G.E. Rectified linear units improve restricted Boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 807–814.

178. Sutskever,I.;Martens,J.;Hinton,G.E. Generating text with recurrent neural networks. In Proceedings of the 28th International Conference on Machine Learning (ICML-11), Bellevue, WA, USA, 28 June–2 July 2011; pp. 1017–1024.

179. Mandic,D.P.;Chambers,J.A. *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability*; Wiley Online Library: Hoboken, NJ, USA, 2001.

180. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166.

181. Hochreiter,S.; Schmidhuber, J.Longshort-termmemory. *Neural Comput.* **1997**, *9*, 1735–1780.

182. Graves,A.;Schmidhuber,J. Frame wise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw.* **2005**, *18*, 602–610.

183. Pascanu, R.; Mikolov, T.; Bengio, Y. On the difficulty of training recurrent neural networks. *ICML* **2013**, *28*, 1310–1318.

184. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv: 1412.3555.

185. Jaderberg,M.; Simonyan,K.; Vedaldi,A.; Zisserman,A. Reading text in the wild with convolutional neural networks. *Int. J. Comput. Vis.* **2016**, *116*, 1–20.

186. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324.

187. Scherer, D.; Müller, A.; Behnke, S. Evaluation of pooling operations in convolutional architectures for object recognition. In Proceedings of the Artificial Neural Networks–ICANN 2010, Thessaloniki, Greece, 15–18 September 2010; pp. 92–101.

188. Johnson,R.;Zhang, T. Effective use of word order for text categorization with convolutional neural networks. *arXiv* **2014**, arXiv:1412.1058.

189. Hinton, G.E. Training products of experts by minimizing contrastive divergence. *Neural Comput.* **2002**, *14*, 1771–1800.

190. Hinton,G.E.;Osindero,S.;Teh,Y.W.A fast learning algorithm for deep belief nets.*Neural Comput.***2006**, *18*, 1527–1554.

191. Mohamed,A.R.;Dahl,G.E.;Hinton,G.A coustic modeling using deep belief networks.*IEEE Trans.Audio Speech Lang. Process.* **2012**, *20*, 14–22.

192. Yang,Z.;Yang,D.;Dyer,C.;He,X.;Smola,A.J.;Hovy,E.H.Hierarchical Attention Networks for Document Classification. In Proceedings of the HLT-NAACL, San Diego, CA, USA, 12–17 June 2016; pp. 1480–1489.

193. Seo,P.H.;Lin,Z.;Cohen,S.;Shen,X.;Han,B.Hierarchical attention networks. *arXiv***2016**, arXiv:1606.02393.

194. Bottou,L.Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 177–186.

195. Tieleman, T.; Hinton, G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA Neural Netw. Mach. Learn.* **2012**, *4*, 26–31.

196. Kingma, D.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.

197. Duchi,J.;Hazan,E.;Singer,Y.Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.* **2011**, *12*, 2121–2159.

198. Zeiler,M.D.ADADELTA:an adaptive learning rate method. *arXiv***2012**,arXiv:1212.5701.

199. Wang,B.;Xu,J.;Li,J.;Hu,C.;Pan,J.S.Scene text recognitional gorithm based on faster RCN N.In Proceedings of the 2017 First International Conference on Electronics Instrumentation & Information Systems (EIIS), Harbin, China, 3–5 June 2017; pp. 1–4.

200. Zhou, C.; Sun, C.; Liu, Z.; Lau, F. A C-LSTM neural network for text classification. *arXiv* **2015**, arXiv:1511.08630.

201. Shwartz-Ziv, R.; Tishby, N. Opening the black box of deep neural networks via information. *arXiv* **2017**, arXiv:1703.00810.

202. Gray,A.;MacDonell,S.Alternative store gression models for estimating software projects.In Proceedings of the IFPUG Fall Conference, Dallas, TX, USA, **1996**.

203. Shrikumar, A.; Greenside, P.; Kundaje, A. Learning important features through propagating activation differences. *arXiv* **2017**, arXiv:1704.02685.

204. Anthes,G.Deep learning comes of age.*Commun.ACM* **2013**,*56*,13–15.

205. Lampinen, A.K.; McClelland, J.L. One-shot and few-shot learning of word embeddings. *arXiv* **2017**, arXiv:1710.10280.

206. Severyn, A.; Moschitti, A. Learning to rank short text pairs with convolutional deep neural networks. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, 9–13 August 2015; pp. 373–382.

207. Gowda,H.S.;Suhil,M.;Guru,D.;Raju,L.N.Semi-supervised text categorization using recursive K-means clustering. In *International Conference on Recent Trends in Image Processing and Pattern Recognition*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 217–227.

208. Kowsari,K.Investigation of Fuzzy find Searching with Golay Code Transformations. PhD thesis,The George Washington University, Department of Computer Science, Washington, DC, USA, 2014.

209. Kowsari,K.;Yammahi,M.;Bari,N.;Vichr,R.;Alsaby,F.;Berkovich,S.Y.Constrution of fuzzy find dictionary using golay coding transformation for searching applications. *arXiv* **2015**, arXiv:1503.06483.

210. Chapelle, O.; Zien, A. Semi-Supervised Classification by Low Density Separation. In Proceedings of the AISTATS, The Savannah Hotel, Barbados, 6–8 January 2005; pp. 57–64.

211. Nigam, K.; McCallum, A.; Mitchell, T. Semi-supervised text classification using EM. In *Semi-Supervised Learning*; MIT Press: Cambridge, MA, USA, 2006; pp. 33–56.

212. Shi,L.;Mihalcea,R.;Tian,M.Cross language text classification by model translation and semi-supervised learning. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Cambridge, MA, USA, 9–11 October 2010; Association for Computational Linguistics: Stroudsburg, PA, USA, 2010; pp. 1057–1067.

213. Zhou, S.; Chen, Q.; Wang, X. Fuzzy deep belief networks for semi-supervised sentiment classification. *Neuro computing* **2014**, *131*, 312–322.

214. Yang,Y.A nevaluation of statistical approaches to text categorization. *Inf. Retr.* **1999**,*1*,69–90.

215. Lever,J.;Krzywinski, M.; Altman,N. Points of significance: Classification evaluation. *Nat.Methods***2016**,*13*, 603–604 .

216. Manning, C.D.; Raghavan, P.; Schütze, H. Matrix decompositions and latent semantic indexing. In *Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2008; pp. 403–417.

217. Tsoumakas,G.;Katakis,I.;Vlahavas,I.Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 667–685.

218. Yonelinas,A.P.;Parks,C.M.Receiver operating characteristics (ROCs) in recognition memory:Areview. *Psychol. Bull.* **2007**, *133*, 800.

219. Japkowicz, N.; Stephen, S. The class imbalance problem: A systematic study. *Intell. Data Anal.* **2002**, *6*, 429–449.

220. Bradley,A.P.The use of thearea under the ROC curvein the evaluation of machine learning algorithms. *Pattern Recogn.* **1997**, *30*, 1145–1159.

221. Hand,D.J.;Till,R.J.A simple generalization of thearea under the ROC curve for multiple class classification problems. *Mach. Learn.* **2001**, *45*, 171–186.

222. Wu,H.C.;Luk,R.W.P.;Wong,K.F.;Kwok,K.L.Interpreting tf-idfterm weights as making relevance decisions. *ACM Trans. Inf. Syst. (TOIS)* **2008**, *26*, 13.

223. Rezaeinia, S.M.; Ghodsi, A.; Rahmani, R. Improving the Accuracy of Pre-trained Word Embeddings for Sentiment Analysis. *arXiv* **2017**, arXiv:1711.08609.

224. Sharma,A.;Paliwal,K.K.Fast principal component analysis using fixed-pointal gorithm.*PatternRecogn. Lett.* **2007**, *28*, 1151–1155.

225. Putthividhya, D.P.; Hu, J. Bootstrapped named entity recognition for product attribute extraction. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Edinburgh, UK, 27–31 July ; Association for Computational Linguistics: Stroudsburg, PA, USA, 2011; pp. 1557–1567.

226. Banerjee,M.*AUtility-Aware Privacy Preserving Framework for Distributed Data Mining with Worst Case Privacy Guarantee*; University of Maryland: Baltimore County, MD, USA, 2011.

227. Chen, J.; Yan, S.; Wong, K.C. Verbal aggression detection on Twitter comments: Convolutional neural network for short-text sentiment analysis. *Neural Comput. Appl.* **2018**, 1–10, doi:10.1007/s00521-018-3442-0.

228. Zhang,X.;Zhao,J.;LeCun,Y.Character-level convolution alnet works for text classification. *Adv.NeuralInf. Process. Syst.* **2015**, *28*, 649–657.

229. Schütze,H.;Manning,C.D.;Raghavan,P.*Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2008; Volume 39.

230. Hoogeveen, D.; Wang, L.; Baldwin, T.; Verspoor, K.M. Web forum retrieval and text analytics: A survey. *Found. Trends® Inf. Retr.* **2018**, *12*, 1–163.

231. Dwivedi,S.K.;Arya,C.Automatic Text Classification in Information retrieval: A Survey. In Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies, Udaipur, India, 4–5 March 2016; p. 131.

232. Jones, K.S. Automatic keyword classification for information retrieval. *Libr. Q.* **1971**, *41*, 338–340, doi:10.1086/619985.

233. O'Riordan,C.;Sorensen,H.Information filtering and retrieval: Anoverview. In Proceedings of the 16th Annual International Conference of the IEEE, Atlanta, GA, USA, 28–31 October 1997; pp. A42–A49.

234. Buckley,C.*Implementation of the SMART Information Retrieval System*; Technical Report; Cornell University: Ithaca, NY, USA, 1985.

235. Pang,B.;Lee,L.Opinion mining and sentiment analysis. *Found. Trends ®Inf. Retr.***2008**,*2*,1–135.

236. Liu, B.; Zhang, L. A survey of opinion mining and sentiment analysis. In *Mining Text Data*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 415–463.

237. Pang,B.;Lee,L.;Vaithyanathan,S. Thumbsup: Sentiment classification using machine learning techniques. In *ACL-02 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2002; Volume 10, pp. 79–86.

238. Aggarwal,C.C.Content-based recommender systems. In *Recommender Systems*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 139–166.

239. Pazzani, M.J.; Billsus, D. Content-based recommendation systems. In *The Adaptive Web*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 325–341.

240. Sumathy,K.;Chidambaram,M.Text mining: Concepts, applications, toolsandissues—Anoverview. *Int.J. Comput. Appl.* **2013**, *80*, 29–32.

241. Heidarysafa,M.; Kowsari,K.; Barnes,L.E.; Brown,D.E.Analysis of Railway Accidents'Narratives Using Deep Learning. In Proceedings of the 2018 17th IEE E International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 17–20 December 2018.

242. Mani,I. *Advances in Automatic Text Summarization*; MIT Press: Cambridge, MA, USA, 1999.

243. Cao, Z.; Li, W.; Li, S.; Wei, F. Improving Multi-Document Summarization via Text Classification. In Proceedings of the AAA I, San Francisco, CA, USA, 4–9 February 2017; pp. 3053–3059.

244. Lauría, E.J.; March, A.D. Combining Bayesian text classification and shrinkage to automate healthcare coding: A data quality analysis. *J. Data Inf. Qual. (JDIQ)* **2011**, *2*, 13.

245. Zhang,J.;Kowsari,K.;Harrison,J.H.;Lobo,J.M.;Barnes,L.E.Patient2Vec:APersonalized Interpretable Deep Representation of the Longitudinal Electronic Health Record. *IEE E Access* **2018**, *6*, 65333–65346, doi:10.1109/ACCESS.2018.2875677.

246. Trieschnigg,D.;Pezik,P.;Lee,V.;DeJong,F.;Kraaij,W.;Rebholz-Schuhmann,D. MeSHUp: Effective MeSH text classification for improved document retrieval. *Bioinformatics* **2009**, *25*, 1412–1418.

247. Ofoghi, B.; Verspoor, K. Textual Emotion Classification: An Interoperability Study on Cross-Genre data sets. In *Australasian Joint Conference on Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 262–273.

248. Pennebaker,J.;Booth,R.;Boyd,R.;Francis,M.*Linguistic Inquiry and Word Count: LIWC2015*; Pennebaker Conglomerates: Austin, TX, USA, 2015. Available online: www.LIWC.net (accessed on Jan 10 2019).

249. Paul,M.J.; Dredze, M. Social Monitoring for Public Health. *Synth. Lect. Inf. Concepts Retr. Serv.* **2017**,*9*,1–183, doi:10.2200/ S00791ED1V01Y201707ICR060.

250. Yu,B.;Kwok,L.Classifying business marketing messages on Facebook. In Proceedings of the Association for Computing Machinery Special Interest Group on Information Retrieval, Bejing, China, 24–28 July 2011.

251. Kang,M.;Ahn,J.;Lee,K.Opinion mining using ensemble text hidden Mark ov models for text classification. *Expert Syst. Appl.* **2018**, *94*, 218–227.

252. Turtle,H.Text retrie valinthelegal world. *Artif.Intell.Law***1995**,*3*,5–54.

253. Bergman,P.;Berman,S.J.*Represent Yourself in Court: Howto Prepare&Trya WinningCase*; Nolo:Berkeley,CA, USA, 2016.

254. Vempala,S.S.(2005).Therandom projection method*(Vol.65). American Mathematical Soc.*