**在文本检索中利用TF-IDF决定词语相关性**

摘要

在本文中，我们研究应用术语频率反向文档频率（TF-IDF）的结果来确定文档语料库中的哪些词可能更有利于在查询中使用。 正如术语所暗示的那样，TF-IDF通过特定文档中单词出现频率的反比例来计算文档中每个单词的值与该单词出现在文档中的百分比。具有高TF-IDF数字的单词意味着与他们出现在其中的文档的关系较强，这表明如果该词出现在查询中，用户可能对该文档感兴趣。 我们提供的证据表明，这种简单的算法能够有效地对可以增强查询检索的相关词进行分类效果。

1.简介

在深入研究我们的实验之前，有必要描述一个文档语料库以及用于解决它的不同方法（包括TF-IDF）的查询检索问题的性质。

1.1查询检索问题

从用户定义的查询中检索数据的任务近年来变得如此普遍和自然，以致有些人可能不会再考虑它。然而，查询检索的这种越来越多的使用需要持续的研究和改进，以便为这个问题提供更好的解决方案。

一般的，查询检索可以被描述为搜索数据集合的任务，即文本文档，数据库，网络等，以用于该数据的特定实例。首先，我们将限制自己在搜索一系列英文文档的过程中。细化的问题随后成为搜索该语料库以查询查询检索系统认为与用户作为查询输入的内容相关的文档的任务。

让我们更正式地描述这个问题。我们有一组文档D，用户输入一个查询q = w1，w2​​，...，wn，查找一系列单词wi。那么我们希望返回D的子集D \*，使得对于每个dєD \*，我们最大化以下概率：

P（d | q，D）（1）

（Berger＆Lafferty，1999）。正如上述符号所表明的，这个问题的许多方法涉及概率和统计，而另一些则提出了基于向量的模型来增强检索的方法。

1.2 Ad-Hoc检索算法

让我们简要回顾一下用于响应查询的其他方法。直观地说，考虑到我们为这个问题提出的正式表示法，统计方法的使用已被证明在回答问题时非常流行和高效。 （Berger＆Lafferty，1999）举例，提出了一个概率框架，在输入查询时加入用户的心态以增强它们的近似值。他们建议用户具有特定的信息需求G，它在实际查询中被近似为字序列q。通过考虑G的这种噪声变换为q并将贝叶斯定理应用于方程（1），它们在返回给定的适当文档时显示出良好的结果。

用于执行查询检索的基于矢量的方法也显示出良好的前景。 （Berry，Dumais＆O'Brien，1994）建议使用称为潜在语义索引（LSI）的流行矩阵算法执行查询检索。本质上，该算法创建了一个缩减维向量空间，用于捕获一组文档的n维表示。当输入查询时，将其数字表示与文档空间中其他文档的余弦距离进行比较，算法返回距离较小的文档。作者的实验结果表明，该算法在查询检索中非常有效，即使这个问题需要对用不同语言编写的文档进行信息检索（Littman＆Keim 1997）。如果满足某些标准，他们建议LSI方法可以扩展到两种以上的语言。我们更详细的检查程序是术语频率逆文件频率（TF-IDF）。这种称重方案可以被归类为统计

程序，尽管其直接结果本质上是确定性的。虽然TF-IDF是一种相对较旧的称重方案，但它简单有效，使其成为其他更新算法的流行起点（Salton＆Buckley，1988）。在本文中，我们将研究TF-IDF对来自LDC联合国并行文本语料库的一组英文文件的行为。本文的目的是研究TF-IDF的行为，优势和弱点，作为未来算法的起点。

2. TF-IDF概述

现在我们将考察一组文档的TF-IDF的结构和实现。 我们将首先介绍该算法的数学背景并检查其相对于每个变量的行为。 然后，我们按照我们的实施情况介绍该算法。

2.1数学框架

我们将在继续之前对TF-IDF进行快速非正式的解释。本质上，TF-IDF通过确定特定文档中词的相对频率与整个文档语料库中该词的反比例进行比较。直观地说，这个计算确定给定单词在特定文档中的相关程度。单个或一小组文件中常见的单词倾向于比常用单词（如文章和介词）具有更高的TF-IDF编号。

实施TF-IDF的正式程序在所有应用程序上都有一些细微差别，但总体方法如下。给定一个文档集合D，一个单词w和一个单独的文档d D，我们计算

wd =ƒw，d \* log（| D | / ww，D）（2），

其中ƒw，d等于w出现在d中的次数，| D |是语料库的大小，ƒw，D等于W出现在D中的文档数量（Salton＆Buckley，1988，Berger等人，2000）。根据ƒw，d，| D |和ƒw，D的值，我们将研究其中最突出的部分，每个单词在这里可能会出现一些不同的情况。

假设| D | ¯w，D，即语料库的大小近似等于D的w的频率。如果对于某个非常小的常数c，1 <log（| D | /ƒw，D）<c，则wd将小于呃，但仍然是积极的。这意味着w在整个语料库中比较常见，但在整个D中仍然具有一定的重要性。例如，如果TF-IDF在新约中检查“耶稣”一词，情况可能如此。与我们更相关的是，联合国文件全文中的“联合国”这个词应该是这个结果。对于极其常见的词语，例如文章，

代词和介词，它们本身在查询中不具有相关含义（除非用户明确地希望包含这种常见单词的文档）。因此，这些常见词语会得到非常低的TF-IDF分数，使得它们在搜索中基本可以忽略不计。

最后，假设ƒw，d很大，ƒw，D很小。那么log（| D | /ƒw，D）将会相当大，所以wd同样会很大。这是我们最感兴趣的情况，因为高wd的单词意味着w是d中的重要词，但在d中不常见。这个术语据说具有很大的区分能力。因此，当查询包含此w时，返回wd很大的文档d很可能会满足用户。

2.2编码TF-IDF

TF-IDF的代码非常简洁。 给定由一组词wi组成的查询q，我们计算wi，d对于每个文档d∈D。最简单的方法是，这可以通过运行文档集合并保持文档集合的运行总和来完成， d和ƒw，D。一旦完成，我们可以根据之前提出的数学框架轻松计算wi。 一旦找到所有wi，d's，我们就返回一个包含文档d的集合D \*，以便我们最大化以下等式：

Σiwi，d（3）。

在启动查询之前，用户或系统可以任意确定D \*的大小。 此外，根据等式（3），文档以递减顺序返回。

这是实施TF-IDF的传统方法。 我们将在后面的章节中讨论该算法的扩展，以及根据我们自己的结果分析TF-IDF。

3.实验

3.1数据收集和格式化

我们在联合国并行文本语料库收集的1400份文件中测试了我们的TF-IDF执行情况。这些文件是从联合国1988年数据库的大量文件中随意收集的。这些文档使用SGML文本格式进行编码，因此我们决定留下格式化标签来解决噪音数据并测试TF-IDF的稳健性。我们通过强制区分大小写来模拟更多噪音。由于某些限制，我们必须将用于执行信息检索的查询数量限制在86个。我们根据等式（3）计算这些查询的TF-IDF权重，然后返回最大化方程（3 ）。退回的文件以降序返回，首先出现重量较高的文件。为了比较我们的结果，我们还同时执行了仅基于术语ƒw，d执行查询检索的蛮力（而非天真）方法。自然地，后一种方法对于查询检索这个较大的问题是直观的缺陷，因为这种方法只会返回文档中不相关的词最多出现的地方（即长文档中有大量的文章和介词可能与查询没有任何关联） 。我们将提供证据表明TF-IDF虽然相对简单，但却比这种原始的方法有了很大的改进。

3.2实验结果

表1。 前8个ƒw，d最高的文件由我们的naïve算法返回，用于查询=“哥伦比亚贩毒”。 高ƒw，d主要来自长篇文章和大量文章和介词。 这些文件的wd得分非常低，而且对于查询而言大多无用。

正如预期的那样，对于简单地返回文件，暴力方法给出的每个查询词都是非常不准确的。 表1中显示的前八个文档中，没有一个与给定查询相关。 在整个344个单词列表中，这种模式是显而易见的，总结在图1中。

图1。 在我们的数据上运行原始的查询检索结果。 请注意，该算法不考虑wd，而是仅基于ƒw，d返回文档。 相关文档零星地散布，所以简单地返回顶层文档就像这个算法一样返回不相关的文档。

表2. TF-IDF返回的前8位wd最高的文件查询=“哥伦比亚毒品贩运”。 题为“国际禁止贩运毒品运动”的最高文件与查询直观相关。

如表2所示，使用TF-IDF检索返回的文档与给定查询高度相关。 前两个文档经常使用查询中的非文章词; 在其他文件中并不常见的词语。 这给出了wd的高和，这反过来又与文件高度相关。 图2显示了我们数据的这种持续模式。

图2.使用TF-IDF检索我们数据的结果。 wd的高值集中在图表的开始处，因此根据这里的主要单词进行查询检索可能会返回相关文档。 这两个额外的图表显示了检索引擎找到的上下界。

显然，TF-IDF比它的对手强大得多。 当检查查询中的单词时，我们发现TF-IDF可以找到经常使用这些单词的文档，并确定它们在文档中是否相关。 TF-IDF的歧视性能力使得检索引擎能够快速找到可能满足用户的相关文档。

4.结论

4.1优点和局限性

我们已经看到，TF-IDF是一种高效且简单的算法，用于将查询中的单词与与该查询相关的文档进行匹配。从收集的数据中，我们看到TF-IDF返回与特定查询高度相关的文档。如果用户要为特定主题输入查询，TF-IDF可以查找包含查询相关信息的文档。此外，编码TF-IDF很直接，因此非常适合形成更复杂算法和查询检索系统的基础（Berger et al，2000）。

尽管TF-IDF强大，但它有其局限性。在同义词方面，请注意TF-IDF不会跳转到单词之间的关系。回到（Berger＆Lafferty，1999），如果用户想要找到关于“牧师”这个词的信息，TF-IDF就不会考虑可能与查询相关的文档，而是使用“reverend” 。在我们自己的实验中，TF-IDF无法将“药物”这个词与其复数“药物”等同起来，将每个词分类为单独的词并略微减少单词的wd值。对于大型文档收藏，这可能会导致问题升级。

4.2进一步研究

由于TF-IDF仅仅是一个主要的基准测试，已经出现了许多算法，使得程序更上一层楼。 （Berger et al，2000）在一篇论文中提出了其中的一部分，其中包括一个称为自适应TF-IDF的TF-IDF版本。该算法包含爬坡和梯度下降以提高性能。他们还提出了一种通过向基准TF-IDF应用统计翻译来在跨语言检索设置中执行TF-IDF的算法。

遗传算法也被用于演变可以匹配或击败TF-IDF方案的程序。 （Oren，2002）采用这种方法来演变。使用遗传编程，变异，交叉和复制的主要思想，本文的作者能够演变出比普通TF-IDF称重方案稍好的程序。尽管作者认为结果不被认为是重要的，但该论文表明学界仍然有兴趣加强简单的TF-IDF方案。

检查我们的数据，我们提高TF-IDF的最简单方法是忽视区分大小写，并将词汇与其词汇派生词和同义词等同起来。未来的研究还可能包括使用TF-IDF在用查询以不同语言编写的文档中执行搜索。增强已经非常强大的TF-IDF算法将增加查询检索系统的成功，查询检索系统很快会成为当前全球信息交换的关键元素。