**R语言自然语言处理：词嵌入（GloVe）**

前文参考：

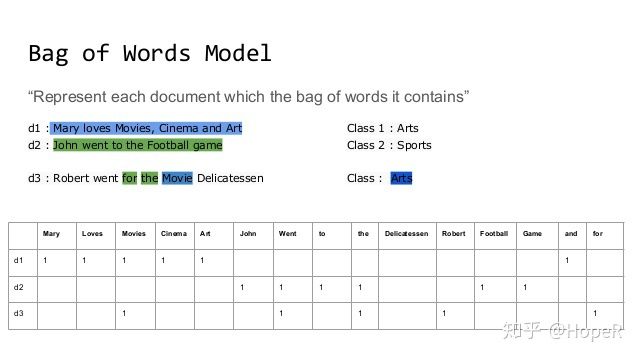
[HopeR：R语言自然语言处理：中文分词](https://zhuanlan.zhihu.com/p/53693842)

[HopeR：R语言自然语言处理：词性标注与命名实体识别](https://zhuanlan.zhihu.com/p/53765293)

[HopeR：R语言自然语言处理：关键词提取（TF-IDF）](https://zhuanlan.zhihu.com/p/53841879)

[HopeR：R语言自然语言处理：关键词提取与文本摘要（TextRank）](https://zhuanlan.zhihu.com/p/54036838)

我们之前讲到的全部都是基于词袋模型（Bag of Words）的方法，比如一个文档某个词出现了多少次，然后就记录下来；进而，我们可以根据逆文本频率指数（IDF）来对其进行优化，但是它依然是基于词袋模型。我们想象一下，一个矩阵，行是文档，列是单词，其中的数值是单词在这个文档中出现的次数，或者是TF-IDF。

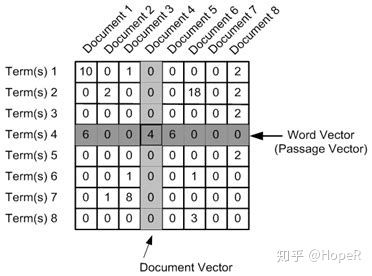


这个模型有两个问题：

1. 如果单词特别多，但是每个词出现的次数又不一定很多的时候，我们得到一个巨大的稀疏矩阵。这样存储效率很低；

2. 这个模型对单词出现的顺序没有任何记录，因此“勇士打败雷霆”和“雷霆打败勇士”这两个短语，在BOW模型中认为意思是完全一样的。

为此，科学家提出了词嵌入模型。



词嵌入（Word Embedding）是文本向量化的一种，概念上而言，它是指把一个维数为所有词的数量的高维空间嵌入到一个维数低得多的连续向量空间中，每个单词或词组被映射为实数域上的向量。比如在上面的图中，我们其实可以把第一列Mary向量化为（1,0,0），但是文档变多的时候效率很低。我们如果使用词嵌入技术，就可以把向量的维度确定在一定范围之内，然后利用一个数值向量来表征这个词，从而来计算不同单词之间的相互关系。利用这个方法，我们有办法找到意思相近的关键词，也可以找到一些词的反义词，还能够计算词与词之间的距离。非常经典的例子是，“国王-男性+女性=女王”。如果我们得到词向量，我们就可以轻易地进行这种计算，然后进行文本的理解。

本来希望从word2vec的方法开始讲实现，但是发现R对word2vec的支持确实不完善，很多教程连数据都无法下载成功，但是有一个包却非常优秀，叫做text2vec。它不能做word2vec，但是却能够很好地实现GloVe。GloVe是基于word2vec的一个词嵌入方法，是斯坦福大学研究者提出的，具体项目见[GloVe项目官网]([https://nlp.stanford.edu/projects/glove/](http://link.zhihu.com/?target=https%3A//nlp.stanford.edu/projects/glove/))。

相关的R代码在[GloVe Word Embeddings](http://link.zhihu.com/?target=http%3A//text2vec.org/glove.html" \t "_blank)，这是目前最新的版本，真实有效，简单快捷。方法的输入，是一段文字，输出是每个词的数值向量。有了这些向量，我们可以做词的聚类，同时也可以用这些词向量来对文档进行分类。此外，fastText也能够对单词进行向量化（<[CRAN - Package fastrtext](http://link.zhihu.com/?target=https%3A//cran.r-project.org/web/packages/fastrtext/index.html)>）。不仅单词可以向量化，整个文档也可以向量化，这个可以使用doc2vec模型，而R中也有相关的实现（<[Doc2Vec function | R Documentation](http://link.zhihu.com/?target=https%3A//www.rdocumentation.org/packages/textTinyR/versions/1.1.2/topics/Doc2Vec)>）。

也就是说，以后如果有需要对文档进行分类，或者对关键词进行分类，词嵌入的技术能够大大帮助我们提高结果的合理性。如果有必要，后期会进行实践操作，目前先介绍到这里。