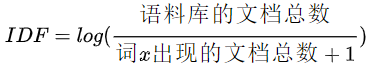
1. 在文本表示领域，除了词嵌入模型（Word Embedding )，经典的模型还有如下三种，词袋模型（Bag of Words) , TF-IDF (Term Frequency-Inverse Document Frequency），主题模型（Topic Model ），请对这三种模型做一个简要的介绍并概括一下优缺点。

词袋模型：

* 定义：不考虑句子中单词的顺序，只考虑词表中单词在这个句子中的出现次数，组成这个句子的向量。
* 优点：可以统计句子中常用词的出现，广泛用于计算机视觉（通过统计图像中特征出现的次数来对图像进行分类）。
* 缺点：未考虑词序，无法反应出一个句子的关键词；在词表大的情况下，句子向量会很稀疏。

TF-IDF：

* 定义：两步处理，第一统计每个词的词频（TF值），第二统计逆文档数（IDF），然后相乘，TF-IDF值越大说明这个词越重要。IDF可以降低像“the, a，of” 这些词的权重。

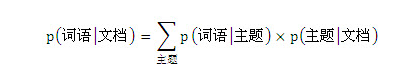


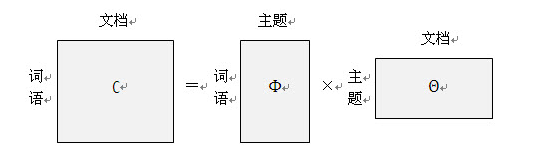
* 优点：学习到了关键词。
* 缺点：未考虑词序，词语间相互独立；严重依赖语料库；

主题模型-LDA： https://blog.csdn.net/jiayalu/article/details/100533184

* 定义：非监督学习的方式对文集的隐含语义结构进行聚类的统计模型。

LDA为例：通过分析主题在词上的概率分布将主题引入文档中，再将文档视为主题的概率分布，从而分析出文档内潜在的主题，进而可以判断文档的相似性。





“文档-词语”矩阵统计了词语在文档中出现的频次，主题-词语”矩阵表示每个主题中每个单词的出现概率；”文档-主题”矩阵表示每个文档中每个主题出现的概率。给定一系列文档，通过对文档进行分词，计算各个文档中每个单词的词频就可以得到左边这边”文档-词语”矩阵。主题模型就是通过左边这个矩阵进行训练，学习出右边两个矩阵。

* 优点：能够分析出文档中的潜在主题；通过主题概率模型能够显著地降低文档特征的维度；可以解决一词多义和多词一义。
* 缺点：不适合短文档的学习。“文档-词语”会很稀疏。这样一个主题包含的词语数和一个词分配给某个主题的次数很难收敛。

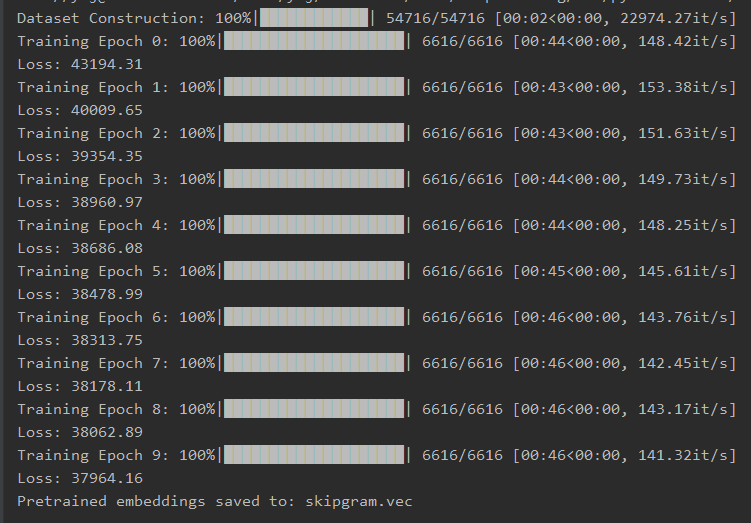
1. 开放问答：提出一种针对低频词的词向量学习改进方案。

* 采用迁移学习获取任务中低频词的词向量
* 参考《一种低频词词向量优化方法及其在短文本分类中的应用》--“计算机科学”

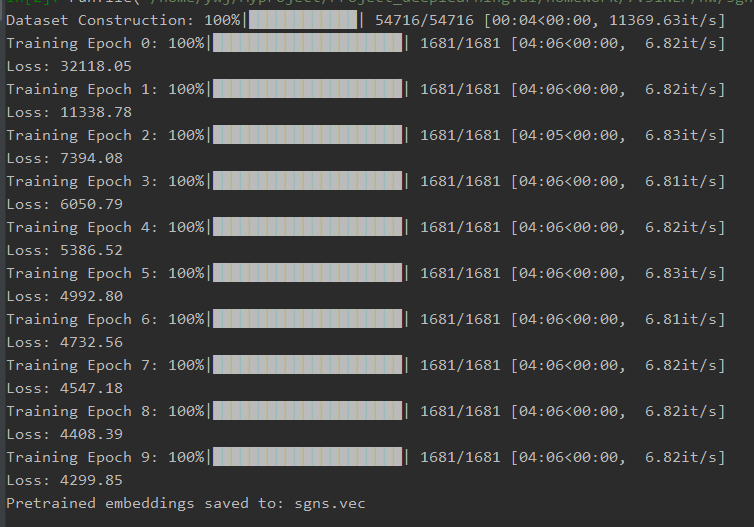
用于下游任务中低频词向量微调：文中提出了一种与下游任务模型无关的低频词词向量更新算法,通过基于K近邻的词向量偏移计算方法,利用通用词向量中与低频词相似的高频词所获得的任务特征信息,来指导低频词的信息更新,从而获得更准确的且适用于当前任务语境的低频词词向量表示。

1. 代码题，给出的文件中，有三种词向量的模型，分别是，CBOW(cbow.py)、Skip-gram(skipgram.py)、基于负采样的Skip-gram模型(sgns.py)，并且还有评估函数（evaluate.py），大家可以挑选任意一个或多个模型进行测试、学习。

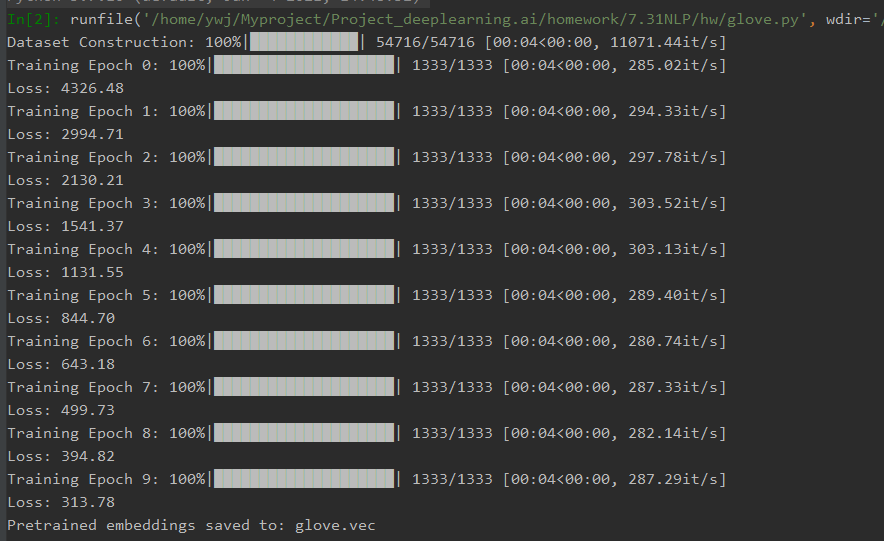
Skip-gram:



Skip-grams:



Glove:



1、Transformer为何使用多头注意力机制？

多头的注意力有助于网络捕捉到子空间的信息，获取更丰富的特征信息。

2、为什么在进行softmax之前需要对attention进行scale操作？

较大的输入会使得softmax的梯度变得很小，反向传播造成梯度消失。

3、在计算attention score的时候如何对padding做mask操作？

将padding的值置为负无穷，在做softmax这些向量为0，消除了padding的影响。

4、Transformer的Decoder端可以做并行化吗？可以的话，如何进行的？

在训练阶段，decoder端可以做并行化，i输入使用的是真实的单词，而不是上一个时间点的输出。并使用seq mask在计算第i个输出的时候，只能看到i之前的输出，而不能看到它后面的，避免数据泄露。

在测试阶段（inference）decoder端不可以做并行化，i的输入要依靠i-1的输出，因为没有真实的标签可以使用。