词嵌入

称为：word embedding或Distributed representation。单个词在预定义的向量空间中被表示为实数向量，每个单词都映射到一个向量。“猫”对应的向量为（0.1 0.2 0.3），“狗”对应的向量为（0.2 0.2 0.4），“爱情”对应的映射为（-0.4 -0.5 -0.2）（本数据仅为示意）。像这种将文本X{x1,x2,x3,x4……xn}映射到多维向量空间Y{y1,y2,y3,y4,y5……yn }，这个映射的过程就叫做词嵌入。

意义：通过计算不同词向量之间夹角余弦值cosine而得出单词之间的相似性。

实现词嵌入的方式：

1）Word2Vec

word2vec主要分为CBOW（Continuous Bag of Words，连续词袋模型）和Skip-Gram(跳词模型)两种模式。

Google论文里真实实现的Word2Vec对模型提出了两种改进思路，即Hierarchical Softmax模型和Negative Sampling模型，借助了分类问题中，使用一连串二分类近似多分类的思想。

Hierarchical(分级的) Softmax是用输出值的霍夫曼编码代替原本的One-Hot向量，用霍夫曼树替代Softmax的计算过程。

Negative Sampling（简称NEG）使用随机采用替代Softmax计算概率，它是另一种更严谨的抽样模型NCE的简化版本。

将这两种算法与前面的两个模型组合，在Google的论文里一共包含了4种Word2Vec的实现。

Hierarchical Softmax CBOW 模型

Hierarchical Softmax Skip-Gram 模型

Negative Sampling CBOW 模型

Negative Sampling Skip-Gram 模型

CBOW在CBOW模型中，通过词的前后词,,,来预测当前词。

Skip-gram模型，通过当前词来预测其前后词,,,。

2）GloVe模型

注：Gensim主题模型包中的word2vec,但通过阅读其源码python版本只实现了skip-gram模型，并且只实现了通过分层softmax方法对其训练，并没有使用negative sampling。java版本分别实现了分层softmax方法对CBOW模型和skip-gram模型进行训练。

窗口大小：为8，即只考虑词前八个和后八个词语。