# Word2Vec学习

经过前期的学习与实践，我主要了解到了Word2Vec作为神经概率语言模型（NLP）的输入，其本身其实是神经概率模型的副产品，是为了通过神经网络学习某个语言模型（CBOW与SKIP-GRAM）而产生的中间结果。会用到两个降低复杂度的近似方法——Hierarchical Softmax或Negative Sampling。两个模型乘以两种方法，一共有四种实现。我首先完成了Word2Vec的实验，然后分析了CBOW和SKIP-GRAM两种模型以及做了Hierarchical Softmax和Negative Sampling的函数推导，主要是在博客，知乎的帮助下完成。

而这一次学习，我因为有了前期的学习与实践，决定从深入理论再到完善实验部分着手继续这一周的Word2Vec学习。

### Tomas Mikolov等人两篇论文学习

[1]Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In Proceedings of Workshop at ICLR, 2013.

[2]Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of NIPS, 2013.

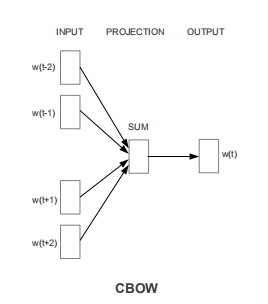
首先为了更加深入了解Word2Vec的发展过程，追根溯源还是Google里Tomas Mikolov等人的两篇论文；查阅资料初步了解后，第一篇Efficient Estimation of Word Representations in Vector Space涉及CBOW和SKIP-GRAM模型这一块，第二篇Distributed Representations of Words and Phrases and their Compositionality涉及Hierarchical Softmax或Negative Sampling的函数实现。

1.Efficient Estimation of Word Representations in Vector Space

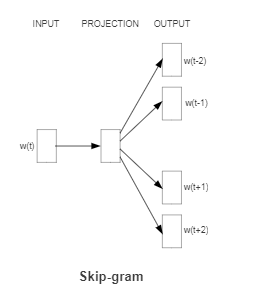
先是第一篇论文，从原因着手如今大部分NLP系统treat words as atomic units，虽然由于其simplicity, robustness and the observation，但是这种简单的放大已经出现了各方面的限制，所以we have to focus on more advanced techniques，并提出了use distributed representations of words这个概念。

（模型框架）然后关于Feedforward Neural Net Language Model (NNLM)，包括input, projection, hidden and output layers，并得出复杂度取决于projection和Hidden层之间。然后指出RNNLM没有映射层. Again, the term *H × V* can be effificiently reduced to *H × log*2(*V* ) by using hierarchical softmax. Most of the complexity then comes from *H × H*. 谈及了层softmax降低复杂度的问题。

（CBOW和SKIP-GRAM）Continuous Bag-of-Words Model，论文给出bag-of-words解释all words get projected into the same position (their vectors are averaged). We call this architecture a bag-of-words model as the order of words in the history does not inflfluence the projection如下图



Continuous Skip-gram ModelThe second architecture is similar to CBOW, but instead of predicting the current word based on the context, it tries to maximize classifification of a word based on another word in the same sentence（这句话解释了SKIP-GRAM与CBOW的不同），增加范围提高生成单词向量的质量，但增加了计算复杂度。同样给出SKIP-GRAM模型如下图。

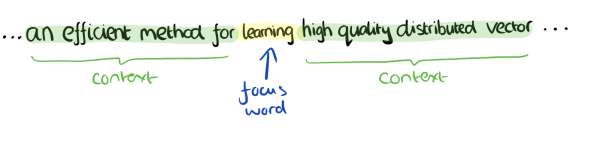


虽然我已经对这两个模型结构比较了解了，但是由于本篇论文比较晦涩难懂还是有些疑惑，所以查阅

CBOW与Skip-Gram模型基础https://www.cnblogs.com/pinard/p/7160330.html

继续去加深我对这部分的理解。

CBOW模型的训练输入是某一个特征词的上下文相关的词对应的词向量，而输出就是这特定的一个词的词向量。

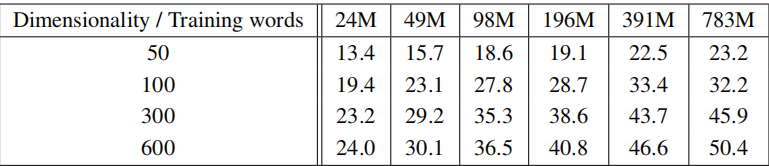


我们的上下文大小取值为4，特定的这个词是"Learning"，也就是我们需要的输出词向量,上下文对应的词有8个，前后各4个，这8个词是我们模型的输入。由于CBOW使用的是词袋模型，因此这8个词都是平等的，也就是不考虑他们和我们关注的词之间的距离大小。我们的输入是8个词向量，输出是所有词的softmax概率，对应的CBOW神经网络模型输入层有8个神经元，输出层有词汇表大小个神经元。隐藏层的神经元个数我们可以自己指定。通过DNN的反向传播算法，我们可以求出DNN模型的参数，同时得到所有的词对应的词向量。这样当我们有新的需求，要求出某8个词对应的最可能的输出中心词时，我们可以通过一次DNN前向传播算法并通过softmax激活函数找到概率最大的词对应的神经元即可。

而对于SKIP-GRAM，与CBOW相反，输入是特定的一个词的词向量，而输出是特定词对应的上下文词向量。还是上面的例子，我们的上下文大小取值为4， 特定的这个词"Learning"是我们的输入，而这8个上下文词是我们的输出。我们的输入是特定词， 输出是softmax概率排前8的8个词，对应的Skip-Gram神经网络模型输入层有1个神经元，输出层有词汇表大小个神经元。隐藏层的神经元个数我们可以自己指定。通过DNN的反向传播算法，我们可以求出DNN模型的参数，同时得到所有的词对应的词向量。这样当我们有新的需求，要求出某1个词对应的最可能的8个上下文词时，我们可以通过一次DNN前向传播算法得到概率大小排前8的softmax概率对应的神经元所对应的词即可。但是这个DNN模型处理过程非常耗时，那它的解决办法在哪？继续看论文。

（结果）Finally, we found that when we train high dimensional word vectors on a large amount of data, the resulting vectors can be used to answer very subtle semantic relationships between words.并提出了可以应用的地方, such as machine translation, information retrieval and question answering systems, and may enable other future applications yet to be invented.

（实验任务）接下来一部分是作者进行的任务，可以借此来对比我的实验并在之后进行改进。To measure quality of the word vectors, we defifine a comprehensive test set that contains fifive types of semantic questions, and nine types of syntactic questions.以及We evaluate the overall accuracy for all question types, and for each question type separately (semantic, syntactic).

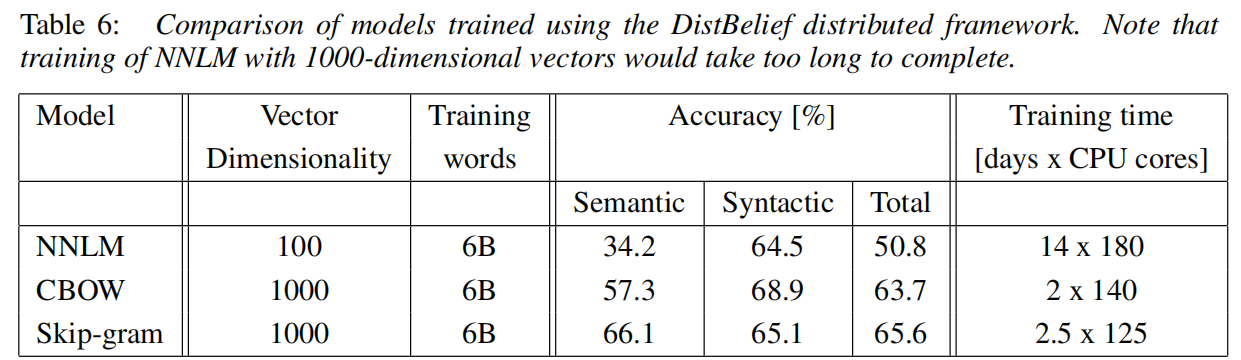


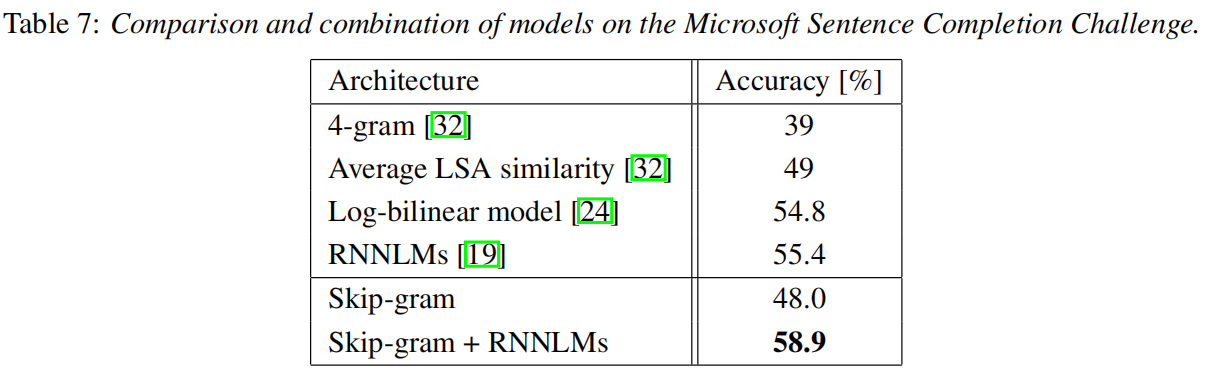
It can be seen that after some point, adding more dimensions or adding more training data provides diminishing improvements.

we used three training epochs with stochastic gradient descent and backpropagation. We chose starting learning rate 0.025 and decreased it linearly, so that it approaches zero at the end of the last training epoch.进行三次训练迭代最后一次迭代近乎为0.我的总结  (1)如果使用更多的数据集和高维度的词向量会有更高的准确率，但是同时也耗费更多的时间。

(2)使用两倍的数据训练效果和迭代三次的效果相似，而且速度还略微加速了

（模型比较与并行训练）





模型优劣势取决于它的精准度以及速度，首先速度来看CBOW和SKIP-GRAM远快于NNLM，而他们俩的精准度与速度差不多，SKIP-GRAM在语义这方面表现比较突出。SKIP-GRAM结合循环神经网络的精准度表现最佳接近60%。

作者还谈及了该模型We believe that word vectors trained on even larger data sets with larger dimensionality will perform signifificantly better, and will enable the development of new innovative applications. Another way to improve accuracy is to provide more than one example of the relationship.以及It is also possible to apply the vector operations to solve different tasks.

（结论）首先作者介绍了本篇论文所涉及的问题，然后指出it is possible to compute very accurate high dimensional word vectors from a much larger data set.以及 it should be possible to train the CBOW and Skip-gram models even on corpora with one trillion words.最后作者对word2vec提出了期望和信心We believe that our comprehensive test set will help the research community to improve the existing techniques for estimating the word vectors. We also expect that high quality word vectors will become an important building block for future NLP applications.

2. Distributed Representations of Words and Phrases and their Compositionality

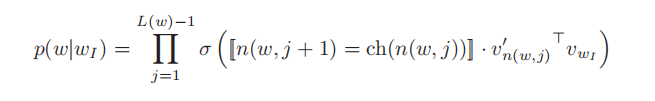
（摘要）

1.In this paper we present several extensions that improve both the quality of the vectors and the training speed.We also describe a simple alternative to the hierarchical softmax called negative sampling.

2.we present a simple method for fifinding phrases in text, and show

that learning good vector representations for millions of phrases is possible.

（Hierarchical Softmax）

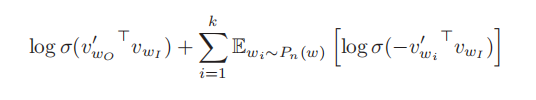


where σ(x) = 1/(1 + exp( x)).

分层softman的优点:it is needed to evaluate only about log2(W) nodes.

而上图这个公式在之前的学习中也是比较熟悉，本质上就是一棵哈夫曼树，关于公式的实际应用也举例子过，还是比较清楚该流程。

（Negative Sampling）



An alternative to the hierarchical softmax is Noise Contrastive Estimation (NCE),

k的合适范围:. Our experiments indicate that values of k in the range 5–20 are useful for small training datasets, while for large datasets the k can be as small as 2–5.

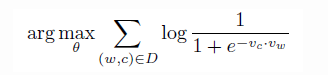
这一块本篇论文讲的比较模糊，所以需要其他文章辅助理解。

<http://blog.sina.com.cn/s/blog_66a6172c0102v1k9.html>

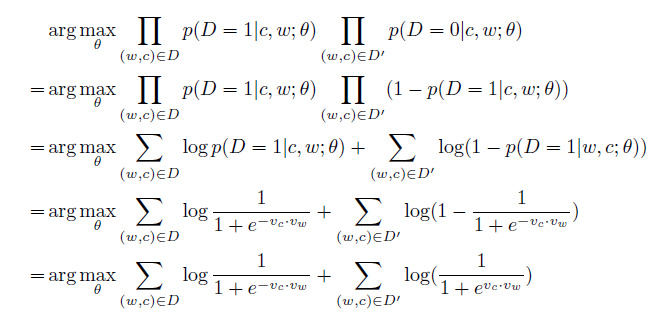
在该篇blog中也有对这两篇论文的理解，翻到negative sampling这一块，它是这样推导的：

对于一个单词、语境组合（w,c），使用p(D=1|w,c)来表示这个组合存在于T中的概率，对应的p(D=0|w,c) = 1-p(D=1|w,c)，表示（w,c）不在T中的概率。假设集合θ是控制p(D=1|w,c)分布的参数，那么此时的目标是寻找参数集合θ来最大化（w,c）存在于T中的概率

经过softmax来量化的目标函数



为了使目标函数最大化，有一个很简单的方法，即使得vc=vw,且vc·vw=K。当K足够大时，可以得到p(D=1|w,c)=1，从而达到目标函数最大化。因此，为了所有的矢量有相同的值，作者生成了一个数据集D'，D'中的所有单词、语境组合都不存在于T中，这些样例被称之为反例（negative examples）,而获得反例的采样方法被称之为反例采集（negative-sampling）。



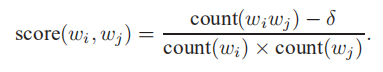
根据联合概率p(D=1|w,c)进行建模。

（Learning Phrases）

To learn vector representation for phrases, we fifirst fifind words that appear frequently together, and infrequently in other contexts.（发现）

This way, we can form many reasonable phrases without greatly increasing the size of the vocabulary;（新的方式）

use a simple data-driven approach

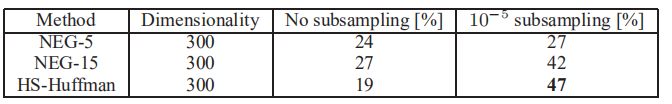


（Phrase Skip-Gram Results）

结果：

1.The results show that while Negative Sampling achieves a respectable accuracy even with k = 5, using k = 15 achieves considerably better performance.

2.while we found the Hierarchical Softmax to achieve lower performance when trained without subsampling, it became the best performing method when we downsampled the frequent words.



这张图就可以看出Hierarchical Softmax和Negative Sampling带来的巨大提升。

This resulted in a model that reached an accuracy of 72%. We achieved lower accuracy 66% when we reduced the size of the training dataset to 6B words, which suggests that the large amount of the training data is crucial.（大量训练数据的重要性）

最后看一下本篇论文的结论：

首先就是通过这两个方法This results in a great improvement in the quality of the learned word and phrase representations.

Another contribution of our paper is the Negative sampling algorithm, which is an extremely simple training method that learns accurate

representations especially for frequent words.

the most crucial decisions that affect the performance are the choice of the model architecture, the size of the vectors, the subsampling rate, and the size of the training window.（什么影响性能）

Our work can thus be seen as complementary to the existing approach that attempts to represent phrases using recursive matrix-vector operations

（作者的工作总结）

### 实验

完成这次实验后上传github

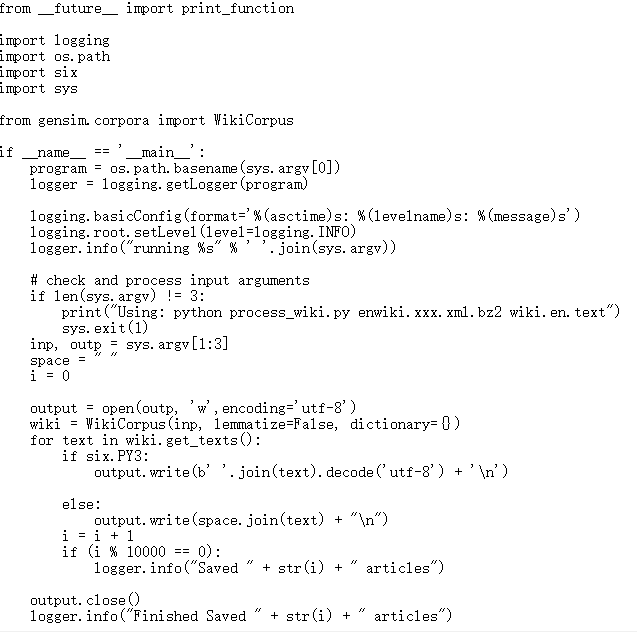
数据方面是没有问题的，继续延用上次维基百科中文语料库并回顾实验过程

<https://dumps.wikimedia.org/enwiki/latest/>

MBFX$A7O7DT9UU$_2AH1K(6

通过process

6A`J5K$2GPE1L$Y@1F7KAUW



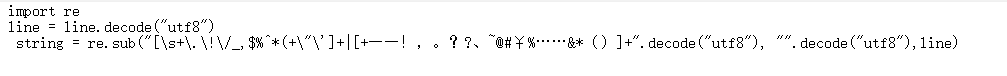
将其转化成我么需要的数据

]35E42W5AOC_2ZW3%WG@MC2

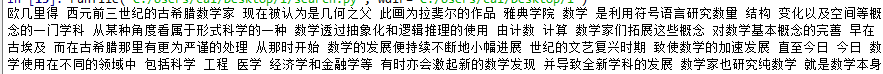
通过opencc工具转化成简体中文

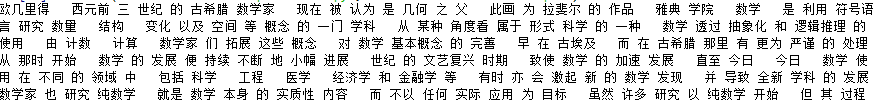
(K%(X3JY5VX7~}W_%VH(D2Q

去掉分隔符及英文

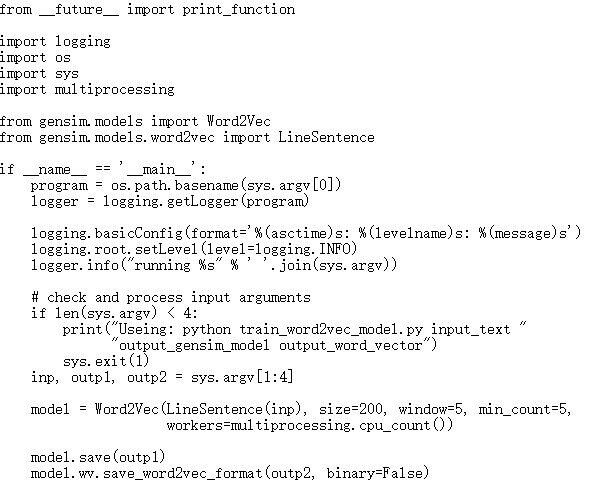


使用jieba分词工具





通过gensim里word2vec训练



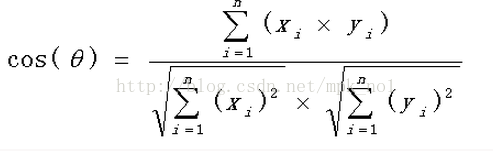
OK获得需要的词向量文件



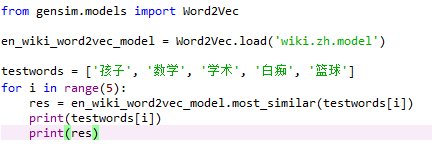
接下来进行一些测试

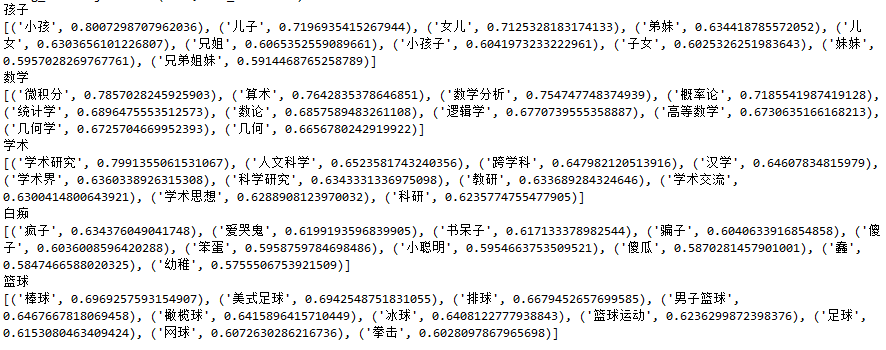
对数据测试参考（gensim里word2vec官方使用文档）https://radimrehurek.com/gensim/models/word2vec.html

1. 求与一个单词相关的单词（窗口设为10）



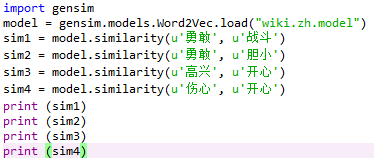
实际上还是求余弦相似度

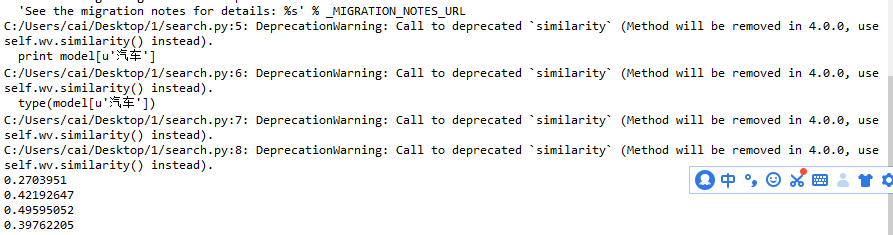




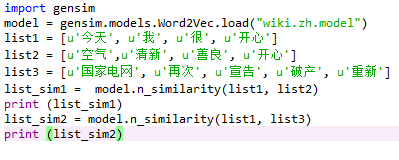
求出的还是上下文最接近的单词有些时候并不是我们观念中的相似。

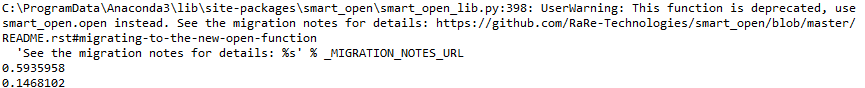
1. 求两个单词（集合）的余弦相似度





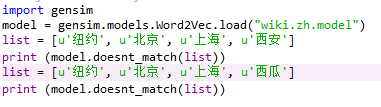
在词性方面表现的非常突出

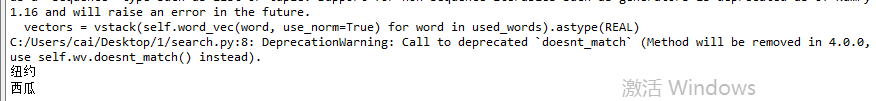




在集合中仍然在词性方面很是敏感

3.选出不同类额词语





总体来说实验还是这几个，还有求向量的大小的model["xxx"]。

通过最近的学习，主要从本质上对word2vec研究更深一些，也通过学习了两篇论文以及一些辅助的blog完善了知识结构。