Report of Deep Learning for Natural Langauge Processing：

Word Vector

Yuqi Shi
1650278787@qq.com

Abstract

This is a Report of Deep Learning for Natural Langauge Processing about word vector.A neural language model Word2Vec is applied to train the word vector---a Chinese corpus is transformed into word vector.To verify the effectiveness of the model Word2Vec,two methods are implemented,including semantic distance between word vectors and words clustering.

Introduction

In natural language processing, words form sentences, and sentences are formed into paragraphs, chapters, and documents, so the first thing to process is to hand over natural language to algorithms in machine learning to process. Words are abstract summaries of human beings, in symbolic form (such as Chinese, English, Latin, etc.), and machines can only accept numerical input, so they need to convert words into numerical form. Word vectors are a way to mathematize words in a language, and word vectors are a way to represent a word as a vector. We all know that words need to be encoded into numeric variables before they are sent to neural network training, and here we use Distributed Representation to encode them.

Distributed Representation was first proposed by Hinton in 1986.The basic idea is that through training, each word in a language is mapped into a short vector of fixed length, all of which constitute a word vector space, and each vector can be regarded as a point in the space, and the grammatical and semantic similarities between them can be judged according to the distance between words. Word2Vec uses this Distributed Representation word vector.

Methodology

The research will be introduced in 3 parts :data preparation,Word2Vec training and K-means clustering.

M1:Data Preparation

The Chinese corpus are 16 novels written by Jin Yiong.Data preprocessing includes file

reading,stop words filter,word segmentation and sampling.After reading the content of the file in UTF8 encoded format, remove all non-Chinese characters in the article, as well as fragments that are not related to the content of the novel, to obtain the form of strings corpus, then use jieba word segmentation for word segmentation, and use Baidu stop word list to filter stop words, and finally return the word segmentation list of the novel.Because the correlation between words needs to be analyzed in the follow-up, the word segmentation list is saved in a txt format file according to 50 words per line for subsequent use.

M2:Word2Vec Training

Word2Vec class in gensim package is used to train the model,and five representative figures in the novels are picked to verify ten words which are more highly relevant to the figure.

```
if __name__ == '__main__':
    test_name = ['郭靖', '杨过', '段誉', '令狐冲', '张无忌']
    model = Word2Vec(sentences=PathLineSentences(DATA_PATH), hs=1, min_count=10, window=5,
                     vector_size=200, sg=0, workers=16, epochs=200)
    for name in test_name:
        print(name)
        for result in model.wv.similar_by_word(name, topn=10):
            print(result[0], '{:.6f}'.format(result[1]))
        print('----------------------')
    model.save('model3.model')
```

The parameters of the model are explained as follows:

sentences: can be a list, the PathLineSentences used here are all files in a folder, and the LineSentence is valid for a single file;

HS: If it is 1, the Hierarchica Softmax technique will be used. If set to 0 (defau·t), negative sampling will be used;

min count: can truncate the dictionary. Words with a word frequency less than min count are discarded, and the default value is 5;

window: indicates the maximum distance between the current word and the predicted word in a sentence;

vector size: refers to the dimension of the feature vector, the default is 100, the larger size requires more training data, but the effect will be better;

Sg: used to set the training algorithm, the default is 0, corresponding to the CBOW algorithm; sg=1 uses skip-gram algorithm;

workers: the number of threads;

epoches: the number of training iterations.

M3:K-means clustering

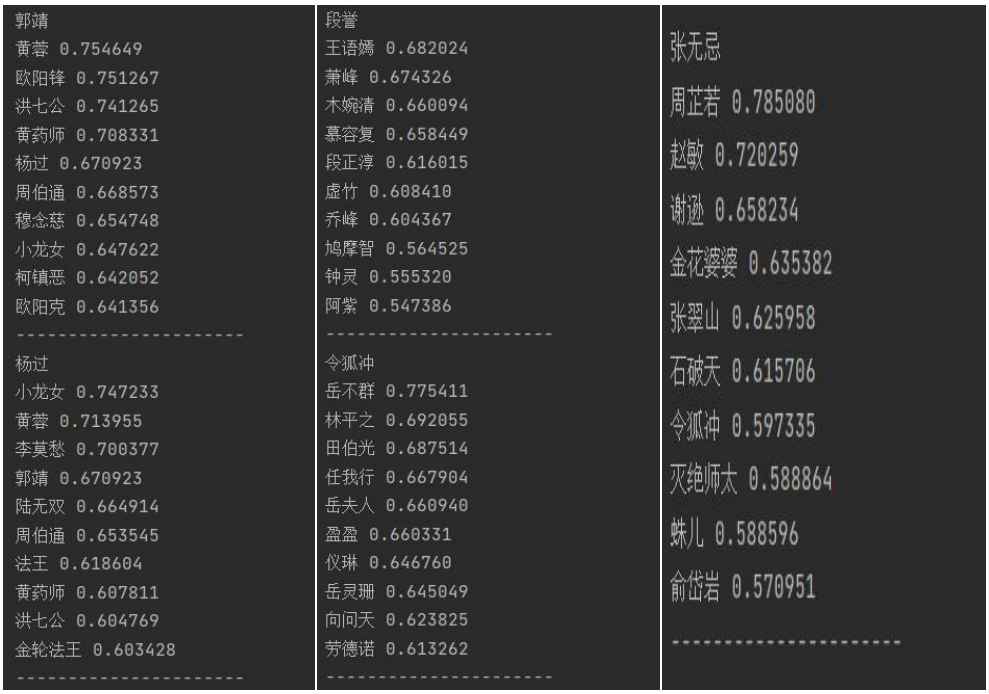
In order to further verify the effectiveness of the model, TSNE was used to reduce the dimensionality of the word vectors in the trained model (which is convenient to show the effect), and the K-means algorithm was used for clustering. The words used in clustering here are representative characters from the five novels. Finally, the effect is displayed with a scatter plot.

Experimental Studies

E1:semantic distance between word vectors

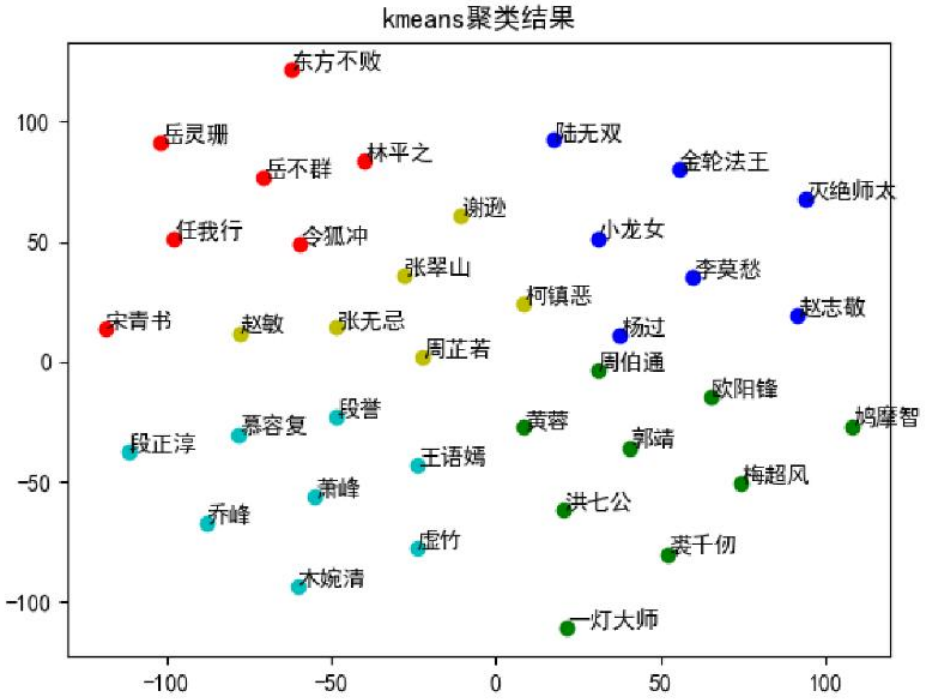
In each of the five novels, a representative figure was selected for the analysis of related words, "The Legend of the Condor Heroes" - Guo Jing, "The Legend of the Condor Heroes" -

Yang Guo, "Dragon Babu" - Duan Yu, "Smiling Proud Jianghu" - Ling Hu Chong, "The Legend of Heavenand Dragon Slayer" - Zhang Wuji, the analysis results are shown as follow.



E2:Clustering Analysis

The results of K-means clustering are as follow:



Conclusions

A Chinese corpus which includes 16 novels of Jin Yong is used as the data and Word2Vec is applied as the model to build and verify the word vector.Results show that the Word2Vec model can help to find the similar semantic words of given word and cluster words into several sets.In a word,word vector is important for NLP.

References

[1] blog.csdn.net/weixin_45314989/article/details/104390725