# Report of Deep Learning for Natural Language Processing

Shixiang Li BY2203043

lishixiang@buaa.edu.cn

## Abstract

This report explores the various representations of word vectors in Natural Language Processing, with a particular focus on three methods: One-Hot Encoding, the Bag-of-Words model, and Distributed Representations. It then provides a detailed introduction to the Word2Vec model, including its two architectures (CBOW and Skip-Gram) and the training process. Finally, the report validates the effectiveness of Word2Vec-based word vectors in tasks such as word similarity and clustering through experiments, and analyzes the advantages and disadvantages of both the CBOW and Skip-Gram models..

## Introduction

Computers cannot directly understand or process human natural language. Therefore, the first step in Natural Language Processing (NLP) is to convert natural text into a numerical form that the computer can "understand." This requires encoding the text into word vectors composed of numbers.

### Common Representations of Word Vectors

(1) One-Hot Encoding

One-Hot Encoding is a method for converting categorical data into binary vectors. Each category is represented by a binary vector of length equal to the number of categories, with only one position set to 1 and all other positions set to 0. For example, given a vocabulary ["cat", "dog", "fish"], the one-hot encoding for these words would be as follows: the encoding for "cat" is [1, 0, 0], for "dog" it is [0, 1, 0], and for "fish" it is [0, 0, 1]. If we have a new sentence, "cat dog dog fish," it can be converted into the following form: [[1, 0, 0], [0, 1, 0], [0, 1, 0], [0, 0, 1]].

The advantage of One-Hot Encoding lies in its ability to map each word to a unique vector, where the vector's dimensionality equals the size of the vocabulary. This ensures the distinctiveness of words, avoiding potential ambiguity issues. However, One-Hot Encoding also has significant limitations. Firstly, since each word vector is unique and orthogonal to others, it fails to capture any semantic similarity or relationship between words. Secondly, as the vocabulary size increases, the dimensionality of the resulting word vectors also grows, which can lead to model overfitting, especially when training data is limited. Furthermore, since the vectors generated by One-Hot Encoding are typically highly sparse, this can negatively impact computational efficiency, particularly when dealing with large-scale datasets. Therefore, while One-Hot Encoding has value in certain contexts, it is generally not the best choice for natural language processing tasks that require capturing semantic relationships between words.

### (2) Bag of Words (BoW)

The Bag of Words (BoW) model is a technique for vectorizing text data. It treats a text as an unordered collection of words, disregarding the order and grammatical structure of the words within the text. In the BoW model, each document is represented as a vector with a dimensionality equal to the size of the vocabulary, where each element of the vector corresponds to the frequency or count of a specific word in the document.

The advantages of the Bag of Words (BoW) model lie in its simplicity and its ability to handle large-scale text data. Since it does not rely on the order of words, it is particularly useful for many classification and clustering tasks. Additionally, the BoW model can be easily combined with other machine learning algorithms. However, the BoW model also has its limitations: First, it loses contextual information. Because the BoW model ignores word order, it cannot capture semantic relationships and contextual information in the text. For example, "I like dogs" and "Dogs like me" would have the same representation in the BoW model, even though their meanings are different. Second, it suffers from the "curse of dimensionality." As the vocabulary size increases, the dimensionality of the feature vector also grows, which can lead to overfitting and reduced computational efficiency. Third, there is sparsity: Most documents only use a small subset of the vocabulary, so the resulting feature vectors are typically highly sparse.

Despite these limitations, the Bag of Words (BoW) model remains a widely used tool, particularly in tasks such as text classification and sentiment analysis. Before the advent of deep learning and word embedding techniques, the BoW model was one of the most commonly used methods for text representation in the field of Natural Language Processing.

### (3) Distributed Representations

Distributed representations are a widely adopted numerical vector representation method in machine learning and Natural Language Processing. Unlike traditional discrete symbolic representations (such as the Bag of Words model), distributed representations encode data into continuous, low-dimensional real-valued vectors, capturing the inherent structure and semantic information of the data. The main characteristics of distributed representations include:

**Continuity:** Distributed representations model data using continuous vector spaces, allowing for a fine-grained quantification of similarities and differences within the data.

**Low Dimensionality:** Distributed representations typically use lower-dimensional vectors, which helps reduce computational complexity and improve the model's generalization ability.

**Semantic Preservation:** Distributed representations are capable of preserving the semantic information of the data, such that elements that are close in the vector space are also semantically similar.

**Generalization Ability:** Distributed representations support vector operations, enabling generalization to unseen data. For instance, through vector addition and subtraction, analogies between words can be inferred.

In the field of Natural Language Processing, word embeddings are a typical application of distributed representations, mapping words to fixed-dimensional vector spaces. Common word embedding techniques include Word2Vec, GloVe, and FastText. These techniques learn vector representations that capture the semantic and usage characteristics of words through training on large-scale corpora. Distributed representations are widely applied in various fields, such as recommendation systems, computer vision, and speech recognition. They provide an effective means of handling high-dimensional data and dimensionality reduction for models. Furthermore, distributed representations demonstrate significant advantages in improving model performance, reducing the risk of overfitting, and enhancing model interpretability.

## Word2Vec

Word2Vec is a computational model that maps each word in the vocabulary to a fixed-dimensional vector. The Word2Vec model was developed by Tomas Mikolov at Google and was released in 2013. The goal of this model is to learn vector representations of words through their context, ensuring that semantically similar words are located close to each other in the vector space.

The Word2Vec model has two architectures: Continuous Bag of Words (CBOW) and Skip-Gram.

**CBOW:** The CBOW model predicts a word based on its context, i.e., the surrounding words. Specifically, it uses the average of the word vectors of the context words to predict the target word. The CBOW model is faster to train on large datasets and performs well with frequent words.

**Skip-Gram:** The Skip-Gram model, in contrast to CBOW, predicts the context from a given word. That is, given a word, the Skip-Gram model tries to predict the words around it. The Skip-Gram model performs better with rare words and complex patterns but requires more training time.

The core idea behind the Word2Vec model is that if two words share similar contexts, their vector representations should also be similar. By training on a large amount of text data, Word2Vec learns the semantic and usage information of words.

The training of the Word2Vec model typically involves the following steps:

**1. Building Training Data:** Word pairs are extracted from text data as training samples. For the CBOW model, the input is the context words, and the target is the center word. For the Skip-Gram model, the input is the center word, and the target is the context words.

**2. Initializing Word Vectors:** Each word in the vocabulary is randomly initialized with a vector.

**3. Training the Model:** Optimization algorithms, such as gradient descent, are used to update the word vectors, allowing the model to better predict the context of words.

**4. Vector Representation:** After training, the vector representations of words can be used for downstream tasks, such as text classification, sentiment analysis, etc.

The Word2Vec model has had a profound impact on the field of Natural Language Processing (NLP). It provides an efficient method for word vector representation and has achieved excellent results in various NLP tasks. Additionally, the Word2Vec model laid the foundation for the development of later word embedding techniques, such as GloVe and FastText.

## Methodology

### M1: Train word vectors based on Word2Vec.

1. Import the corpus file.

2. Remove stop words from the corpus data.

3. Perform word segmentation using jieba.

4. Save the segmented corpus results.

5. Load the saved corpus and train the Word2Vec model.

6. Save the trained CBOW model and Skip-Gram model.

**M2: Display word similarity.**

1. Load the trained model.

2. Specify a particular word and display the 5 most similar words to it.

**M3: Validate the effectiveness of word vectors through clustering.**

1. From the corpus consisting of 16 novels, select words that appear more than 50 times as high-frequency words, and filter out stop words from the high-frequency words.

2. For all remaining high-frequency words, obtain their word vectors using the trained skip-gram model.

3. Use the K-means clustering method to cluster these word vectors, setting the number of clusters to 16.

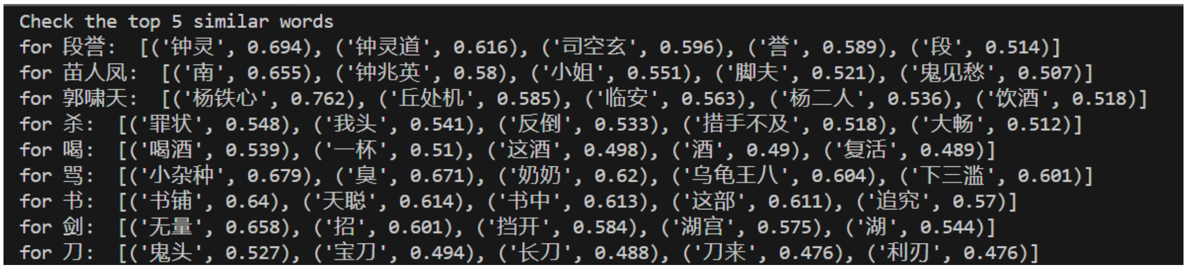
4. Visualize the clustering results using the t-SNE method.

## Experimental Studies

### E1: Word Similarity Query

Firstly, most similar words of given word are evaluated. For example, we select 9 words to test, including names, verbs, and nouns ("段誉","苗人凤", "郭啸天","杀","喝","骂","书","剑","刀"). The results are shown below in **Figure 1**.

From **Figure 1**, it can be found that for experts familiar with the content and characteristics of wuxia novels, the given word does indeed have a strong association with the output words semantically. For example, "段誉" is highly related with '钟灵',"喝" is related with "酒", "书" is related with "书铺". Thus the effectiveness of the word2vec can be verified.

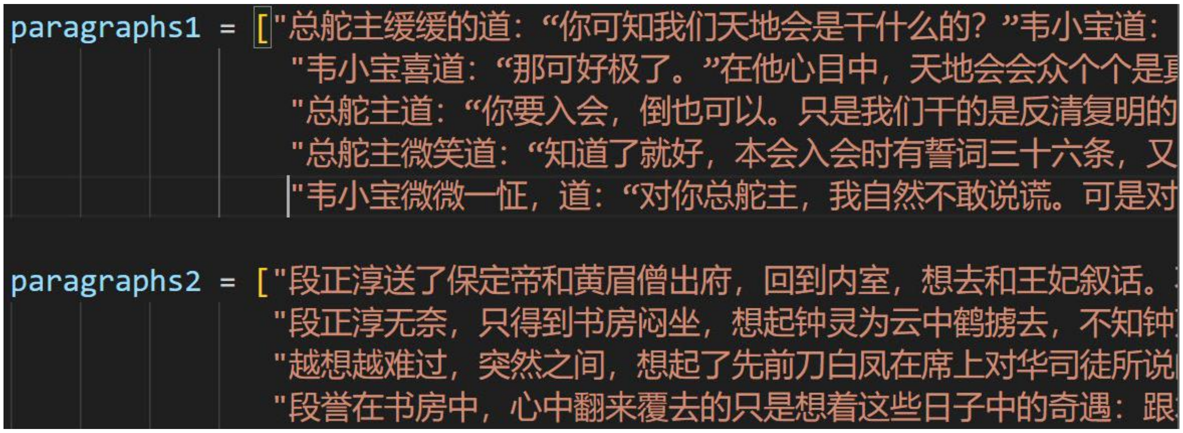


```
Check the top 5 similar words
for 段誉:  [('钟灵', 0.694), ('钟灵道', 0.616), ('司空玄', 0.596), ('誉', 0.589), ('段', 0.514)]
for 苗人凤:  [('南', 0.655), ('钟兆英', 0.58), ('小姐', 0.551), ('脚夫', 0.521), ('鬼见愁', 0.507)]
for 郭啸天:  [('杨铁心', 0.762), ('丘处机', 0.585), ('临安', 0.563), ('杨二人', 0.536), ('饮酒', 0.518)]
for 杀:  [('罪状', 0.548), ('我头', 0.541), ('反倒', 0.533), ('措手不及', 0.518), ('大畅', 0.512)]
for 喝:  [('喝酒', 0.539), ('一杯', 0.51), ('这酒', 0.498), ('酒', 0.49), ('复活', 0.489)]
for 骂:  [('小杂种', 0.679), ('臭', 0.671), ('奶奶', 0.62), ('乌龟王八', 0.604), ('下三滥', 0.601)]
for 书:  [('书铺', 0.64), ('天聪', 0.614), ('书中', 0.613), ('这部', 0.611), ('追究', 0.57)]
for 剑:  [('无量', 0.658), ('招', 0.601), ('挡开', 0.584), ('湖宫', 0.575), ('湖', 0.544)]
for 刀:  [('鬼头', 0.527), ('宝刀', 0.494), ('长刀', 0.488), ('刀来', 0.476), ('利刃', 0.476)]
```

**Figure 1 The Most Similar 5 Words of Given Word**

### E2: Paragraph Similarity Query

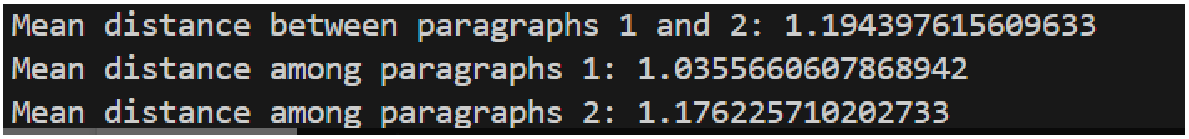
Secondly, this experiment measures the distance of several paragraphs extracted from different texts. For example, several paragraphs in 《鹿鼎记》 and 《天龙八部》are selected, shown in **Figure 2**. Then, the Word Mover's Distance between the two paragraph lists and among each are calculated. The result is shown in **Figure 3**. From **Figure 3**, it can be found that the mean distance between the two parahgraphs is larger than the distance among each parahgraphs. It indicates that by using the word2vec model, the paragraphs from the same text may have a relatively low distance.



```
paragraphs1 = ["总舵主缓缓的道：“你可知我们天地会是干什么的？”韦小宝道：
                "韦小宝喜道：“那可好极了。”在他心目中，天地会会众个个是
                "总舵主道：“你要入会，倒也可以。只是我们干的是反清复明的
                "总舵主微笑道：“知道了就好，本会入会时有誓词三十六条，又
                "韦小宝微微一怔，道：“对你总舵主，我自然不敢说谎。可是对

paragraphs2 = ["段正淳送了保定帝和黄眉僧出府，回到内室，想去和王妃叙话。
               "段正淳无奈，只得到书房闷坐，想起钟灵为云中鹤掳去，不知钟
               "越想越难过，突然之间，想起了先前刀白凤在席上对华司徒所说
               "段誉在书房中，心中翻来覆去的只是想着这些日子中的奇遇：跟
```

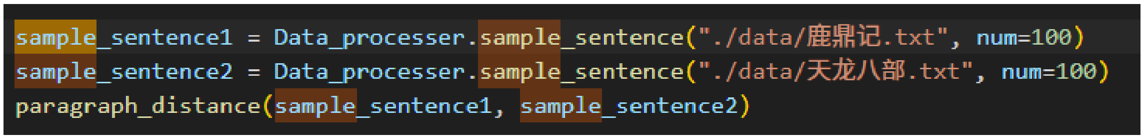
**Figure 2 Two paragraphs list selected from two novels**



```
Mean distance between paragraphs 1 and 2: 1.194397615609633
Mean distance among paragraphs 1: 1.0355660607868942
Mean distance among paragraphs 2: 1.176225710202733
```

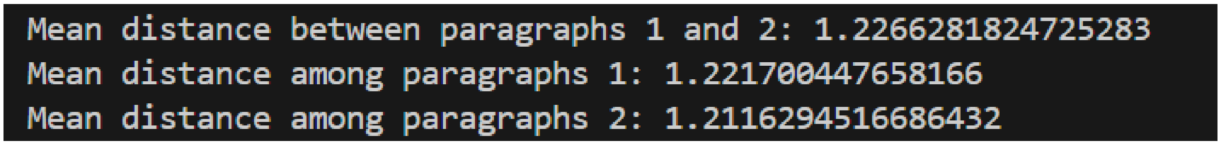
**Figure 3 The distances between the two paragraph lists and among each**

Besides, this experiment is also conducted on the random selected paragraphs from two texts. From each text, 100 random lines are selected, and the Word Mover's Distance between the two paragraph lists and among each are calculated. From **Figure 5**, it can also be found that the mean distance between the two parahgraphs is larger than the distance among each parahgraphs, but the distance is shorter. It can be concluded that low semantic association can cause a larger distance, but those paragraphs from the same text still has more semantic association. Thus, the effectiveness of the word2vec can be verified.



```
sample_sentence1 = Data_processer.sample_sentence("./data/鹿鼎记.txt", num=100)
sample_sentence2 = Data_processer.sample_sentence("./data/天龙八部.txt", num=100)
paragraph_distance(sample_sentence1, sample_sentence2)
```

**Figure 4 Two random paragraphs list selected from two novels**

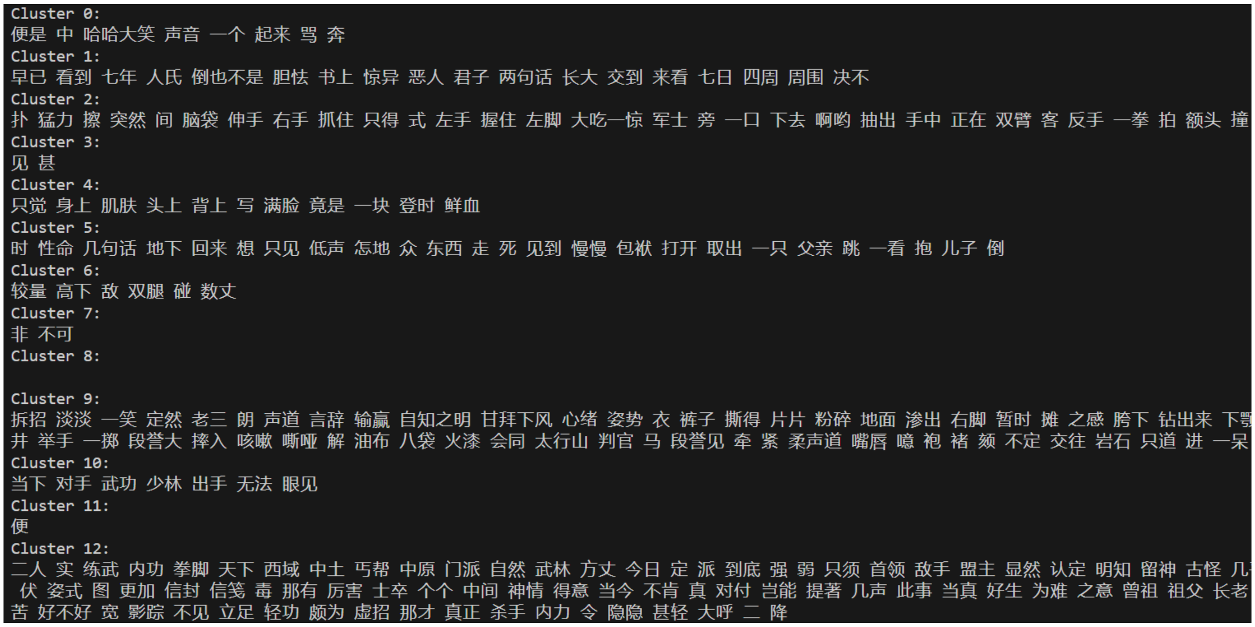


```
Mean distance between paragraphs 1 and 2: 1.2266281824725283
Mean distance among paragraphs 1: 1.221700447658166
Mean distance among paragraphs 2: 1.2116294516686432
```

**Figure 5 The distances between the two random paragraph lists and among each**

## E3: Word clustering

Here word clustering method is also applied to verify the effectiveness of word2vec model. It is conducted on the random selected paragraphs from text《天龙八部》. The result is shown in **Figure 6**. From **Figure 6**, we can see that there are some clusters are meaningful, such as cluster 2:(“扑”, “抓住”, “伸手”, “抓住”, “拍”...), cluster 4:(“肌肤”, “头上”, “背上”, “鲜血”,...), cluster 12:(“练武”, “西域”, “中土”, “门派”, “武林”, “方丈”...). Those words in each cluster are semantically related, which indicates the effectiveness of the word2vec model.



```
Cluster 0:
便是 中 哈哈大笑 声音 一个 起来 骂 奔
Cluster 1:
早已 看到 七年 人氏 倒也不是 胆怯 书上 惊异 恶人 君子 两句话 长大 交到 来看 七日 四周 周围 决不
Cluster 2:
扑 猛力 擦 突然 间 脑袋 伸手 右手 抓住 只得 式 左手 握住 左脚 大吃一惊 军士 旁 一口 下去 啊哟 抽出 手中 正在 双臂 客 反手 一拳 拍 额头 撞
Cluster 3:
见 甚
Cluster 4:
只觉 身上 肌肤 头上 背上 写 满脸 竟是 一块 登时 鲜血
Cluster 5:
时 性命 几句话 地下 回来 想 只见 低声 怎地 众 东西 走 死 见到 慢慢 包袱 打开 取出 一只 父亲 跳 一看 抱 儿子 倒
Cluster 6:
较量 高下 敌 双腿 碰 数丈
Cluster 7:
非 不可
Cluster 8:

Cluster 9:
拆招 淡淡 一笑 定然 老三 朗 声道 言辞 输赢 自知之明 甘拜下风 心绪 姿势 衣 裤子 撕得 片片 粉碎 地面 渗出 右脚 暂时 摊 之感 胯下 钻出来 下
井 举手 一掷 段誉大 摔入 咳嗽 嘶哑 解 油布 八袋 火漆 会同 太行山 判官 马 段誉见 牵 紧 柔声道 嘴唇 噫 袍 褚 颏 不定 交往 岩石 只道 进 一呆
Cluster 10:
当下 对手 武功 少林 出手 无法 眼见
Cluster 11:
便
Cluster 12:
二人 实 练武 内功 拳脚 天下 西域 中土 丐帮 中原 门派 自然 武林 方丈 今日 定 派 到底 强 弱 只须 首领 敌手 盟主 显然 认定 明知 留神 古怪 几
 伏 姿式 图 更加 信封 信笺 毒 那有 厉害 士卒 个个 中间 神情 得意 当今 不肯 真 对付 岂能 提著 几声 此事 当真 好生 为难 之意 曾祖 祖父 长老
苦 好不好 宽 影踪 不见 立足 轻功 颇为 虚招 那才 真正 杀手 内力 令 隐隐 甚轻 大呼 二 降
```

**Figure 6 Word clustering result**

## Conclusion

In this study, a Word2Vec model on Chinese corpus are built, trained and verified. From the experimental studies, in can be concluded that the Word2Vec model can help find the most similar words of given word, calculate and compare the distances between different paragraphs attained from different texts, and cluster words into several sets by  the vector representations. In a word, Word2Vec model can capture the semantics and contextual relationships of vocabulary , and word vector is helpful in many NLP tasks.

## References

[1] https://paddlepedia.readthedocs.io/en/latest/tutorials/sequence_model/word_representation/word2vec.html