# Word Vectors based on Jin Yong Novels

Hao Lu

lh2002@buaa.edu.cn

## Abstract

In this study, we trained word embeddings using a corpus of Jin Yong's novels with two distinct methods: Word2Vec (CBOW) and a single-layer LSTM. We analyzed the resulting word vectors using t-SNE for visualization and cosine similarity for semantic evaluation. The experimental results demonstrate that both models capture meaningful semantic relationships, with LSTM showing stronger contextual sensitivity. Our findings provide insight into the effectiveness of different embedding strategies on literary text. **Interestingly, the sequential data organization used in LSTM models can be seen as a precursor to modern Transformer-based large language models (LLMs). Conceptually, LLMs replace the recurrent LSTM layer with stacked Transformer decoder layers, enabling more efficient parallel computation and better long-range dependency modeling.**

## Introduction

This experiment aims to explore the fundamental properties of word embeddings and to investigate different methods for obtaining them.We use the corpus of Jin Yong's novels to train word embeddings using two different approaches: the Word2Vec model and a single-layer LSTM. By applying these two methods, we obtain distinct word representations. To analyze and compare the characteristics of the learned embeddings, we employ t-SNE to visualize the word vectors generated by both models.

## Methodology

### Data Preprocess

We use Jieba for text segmentation, splitting the text into paragraphs based on newline characters. To accelerate training and improve data quality, we remove all paragraphs that are shorter than 500 characters.

## Params

In this paper, we trained word embeddings using both the Word2Vec and LSTM models. For the Word2Vec model, the following parameters were set: `vector_size=100`, `window=5`, `min_count=5`. For the LSTM model, we used a single-layer LSTM architecture, with both the word embedding dimension and hidden layer dimension set to 128.

## Method Details

### Word2Vec

Word2Vec is a model used for generating word embeddings. It learns to map words into a continuous vector space using a neural network, such that semantic information of words is represented through the distances and relationships between vectors. There are two main training methods in Word2Vec: CBOW (Continuous Bag of Words) and Skip-gram.

**CBOW (Continuous Bag of Words)**

The CBOW model predicts the target word based on its context words. Given a context window of size k, CBOW tries to predict the target word in the middle of the context. For example, given the context words ”The cat is on the”, CBOW will predict the target word ”mat”.

**Skip-gram**

The Skip-gram model, in contrast to CBOW, predicts context words based on a target word. For instance, given the target word ”cat”, Skip-gram will try to predict the context words ”The”, ”is”, ”on”, ”the”.

In this experiment, we chose the CBOW method to train the word embeddings.

### LSTM

We use LSTM (Long Short-Term Memory) networks to train word embeddings by leveraging sequential context. The process can be summarized as follows:

**Data Preprocessing:**

Text data is tokenized, and a vocabulary is created. Sequences are padded or truncated to ensure uniform input length.

**Model Architecture:**

Embedding Layer: Maps each word to a dense vector representation.

LSTM Layer: Captures the sequential dependencies in the text.

Output Layer: Predicts the next word in the sequence using a softmax activation.

**Training:**

The model is trained to minimize the prediction error, using cross-entropy loss. Word vectors are updated based on the context learned from the sequences.

Word Embeddings: After training, the word embeddings are extracted from the embedding layer and used for downstream tasks.

This approach enables the model to learn context-sensitive word representations that capture both syntactic and semantic relationships in the text.

# Experimental Studies

## Word2Vec Visualization

We used the t-SNE method to perform dimensionality reduction on the word embeddings.

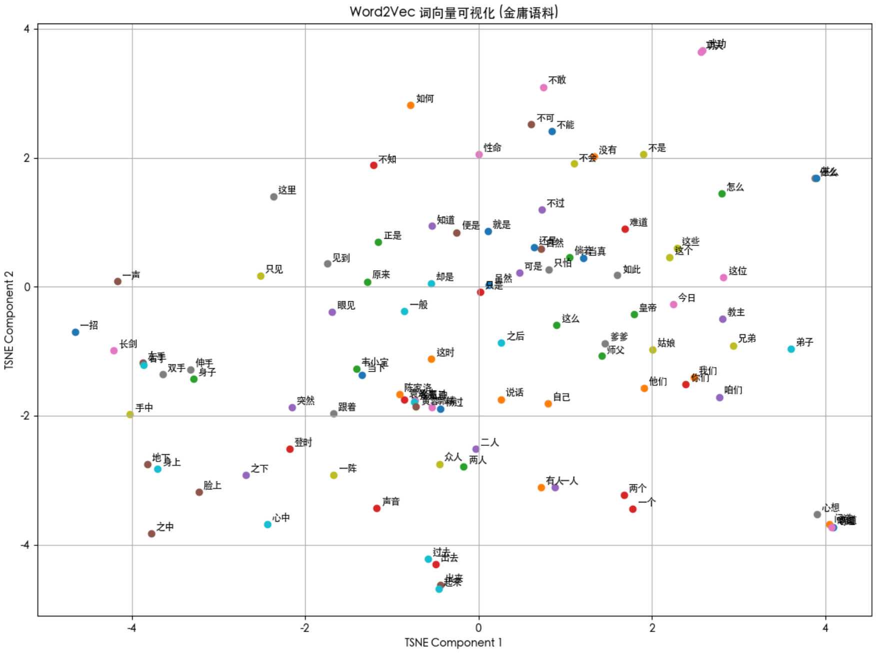


Figure 1: Word2Vec Visualization

## LSTM Visualization

We selected a word (meaning ”two”) for the word vector distance analysis. Using cosine similarity, we computed the top 3 most similar words from the first 500 words in the vocabulary. The results are as follows, demonstrating that the learned word embeddings can effectively capture semantic relationships between words to a certain extent.

# Conclusions

This experiment compared Word2Vec and LSTM approaches for training word embeddings on the Jin Yong novel corpus. Word2Vec with CBOW efficiently

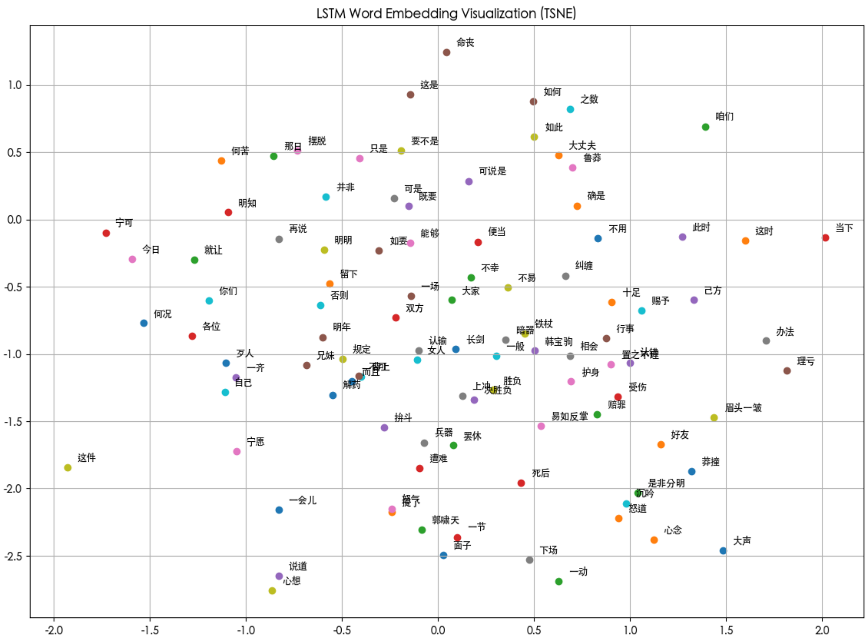
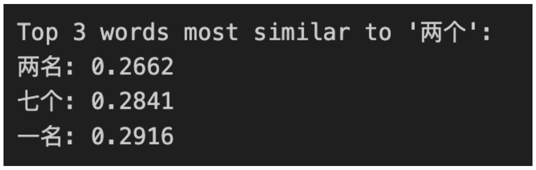


Figure 2: LSTM Visualization
Similarly, we used t-SNE to visualize the word embeddings learned by the LSTM model.

captured local context and word associations, while LSTM embeddings reflected deeper sequential and contextual relationships. Visualization and similarity analysis confirmed the models' ability to encode semantic meaning. These results highlight the importance of model choice in downstream tasks involving semantic understanding and classification in literary or long-form texts.



```
Top 3 words most similar to '两个':
两名: 0.2662
七个: 0.2841
一名: 0.2916
```

Figure 3: Top 3 word