Technology Review

Zerui Tian

zeruit2

Word Embedding & Sentence Embedding

Introduction

Word embedding is a very important technology in natural language processing. When we first process words, we will use one-hot encoding, but the dimension of such words will be very high and the resulting word matrix is actually very sparse. And word embedding can embed a high-dimensional space into a continuous vector space with a much lower dimension. This is very beneficial for us to analyze words, which is why I want to study this field.

Embedding Layer

Embedding layers are generally related to technologies and concepts related to neural network models. In many cases, when processing text data, we will perform one-hot encoding on words, and then specify the dimension of the vector space. Generally, the embedding layer will be used in the front part of the neural network to train the word to a word vector.

GloVe

The full name of the GloVe method is Global Vectors for Word Representation. The principle of its realization is mainly based on the word representation tool of global word frequency statistics. There are generally three steps to its realization:

1. Build a co-occurrence matrix from the corpus. In the matrix, X_{ij} represents the number of co-occurrences of the word i and word j in a context window of a certain size. And a decay function is proposed to calculate the weights.
2. Use the following formula to construct an approximate relationship between word vectors and co-occurrence matrices: $\omega_i^T \omega_j + b_i + b_j = \mathbf{log}(X_{ij})$. Here ω_i and ω_j are the word vectors we want and b_i is the biased term.
3. Finally, we can construct the loss function: $J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b_j} - \log\left(X_{ij}\right)\right)^2$. Here $f(X_{ij})$ is the weight function. We want words that occur frequently to be given more weight than words that occur rarely together.

We only need to train according to the gradient descent method to get the trained word vector.

Word2Vec

The training model of Word2Vec is essentially a neural network structure with hidden layers. What we want to get is the weight from the input layer to the hidden layer, this weight is actually the word vector we want to get in the end. And there are two models: CBOW and Skip-gram. CBOW uses the context of the center word as input to predict the center word, while Skip-Gram uses the center word to predict the context word. Among them, CBOW is generally suitable for small datasets, while Skip-Gram performs better in large corpora.

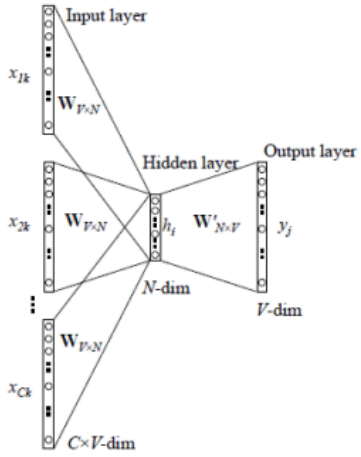

Figure 2: Continuous bag-of-word model

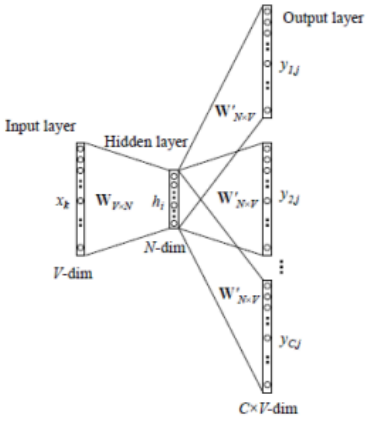

Figure 3: The skip-gram model.

Two very important methods in the algorithm are hierarchical softmax and negative sampling. Hierarchical softmax uses the Huffman tree to replace the matrix from the hidden layer to the output layer. Negative sampling methods allow us to use a smaller number of negative samples to update the sample weights. Both of these methods can reduce the training overhead and speed up the training of the model.

ELMo

ELMo is short for Embedding from Language Models. Word embeddings cannot handle polysemous words, so many times the results obtained are not accurate or can not be widely used. ELMo is a word embedding method that combines context words. It first obtains a pre-trained word embedding method and then adjusts the word embedding vector of the word according to the meaning of the context word. Due to the combination of context, the problem of polysemy can be solved to a certain extent.

BERT

The full name of BERT is Bidirectional Encoder Representation from Transformers. It is a very powerful method to build a Masked language model, which is to randomly cover or replace words in a sentence, and let the model predict the masked or replaced parts through context. And BERT also has a task of Next Sentence Prediction. Its Embedding is obtained by summing the three Embeddings. Among them, Token Embeddings is the word vector, Segment Embeddings is used to distinguish two kinds of sentences, and Position Embeddings is used to represent the word vector of words in different positions.

Sentence Embedding

For sentence embeddings, a relatively simple method is to first obtain the vector of word embeddings, and then simply average these vectors to obtain sentence embeddings. The TF-IDF method, we learned in class can also embed sentences. But we can actually combine these two methods, we can calculate the weighted average of the word vectors in the sentence, and then combine the embedding vectors of many sentences, and then do principal component analysis on it. All sentence vectors are then subtracted by their projections on their respective first principal component vectors. The weighted average method is that we assume that high-frequency words are not helpful for the unique meaning of sentences, so we add weights. Such a method actually removes information that is not helpful for understanding the meaning of the sentence, and better reflects the information we want.

Conclusion

We can find that there are many methods for word embedding, and sometimes different word embedding or sentence embedding methods may have different effects for different tasks. In fact, there are many other text embedding methods including fasttext and so on. Since text embeddings have important implications for text mining or natural language processing, I think it makes sense to do a technologyl review in this area. There may be more and more mature text embedding methods based on BERT in the future, and I am very excited to learn them.

Reference

1.Arora S, Liang Y, Ma T. A simple but tough-to-beat baseline for sentence embeddings[C]//International conference on learning representations. 2017.

2.Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.

3.Pennington J, Socher R, Manning C D. Glove: Global vectors for word representation[C]//Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014: 1532-1543.

4.Sarzynska-Wawer J, Wawer A, Pawlak A, et al. Detecting formal thought disorder by deep contextualized word representations[J]. Psychiatry Research, 2021, 304: 114135.

5.Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.