Word2Vec

1

paper we present several extensions that improve both the quality of the vectors and the training speed.

1

문제

An inherent limitation of word representations is their indifference to word order and their inability to represent idiomatic phrases. For example, the meanings of “Canada” and “Air” cannot be easily combined to obtain “Air Canada”. Motivated by this example, we present a simple method for finding phrases in text, and show that learning good vector representations for millions of phrases is possible

1 indifference to word order.

2 inability to represent idiomatic phrase

2

Skip-gram model

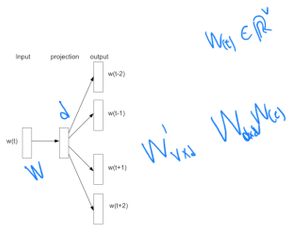

Figure 1: The Skip-gram model architecture. The training objective is to learn word vector representations that are good at predicting the nearby words.

2

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\neq 0}\log p(w_{t+j}|w_t)$$

Maximize average log probability

$w_1, w_2 \cdots w_T$: given sequence of word

C : size of context

Softmax

$$p(W_O|W_I) = \frac{\exp(v'^{\top}_{w_O} v_{w_I})}{\sum_{w=1}^{W}\exp(v'^{\top}_{w} v_{w_I})}$$

W : # of word.

3

Hierarchical Softmax

$$p(w|w_I) = \prod_{j=1}^{L(w)-1}\sigma\left([\![n(w,j+1)=\mathrm{ch}(n(w,j))]\!]\cdot v'^{\top}_{n(w,j)} v_{w_I}\right) \quad (3)$$

Each word w.

each word w can be reached by an appropriate path from the root of the tree. Let n(w, j) be the j-th node on the path from the root to w, and let L(w) be the length of this path, so n(w, 1) = root and n(w, L(w)) = w. In addition, for any inner node n, let ch(n) be an arbitrary fixed child of n and let [[x]] be 1 if x is true and -1 otherwise.

Negative Sampling

$$\log\sigma(v'^{\top}_{w_O} v_{w_I}) + \sum_{i=1}^{k}\mathbb{E}_{w_i\sim P_n(w)}\left[\log\sigma(-v'^{\top}_{w_i} v_{w_I})\right] \quad (4)$$

Negative sampling.

4

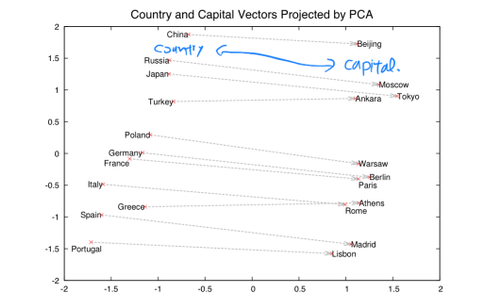

Figure 2: Two-dimensional PCA projection of the 1000-dimensional Skip-gram vectors of countries and their capital cities. The figure illustrates ability of the model to automatically organize concepts and learn implicitly the relationships between them, as during the training we did not provide any supervised information about what a capital city means.

4

To counter the imbalance between the rare and frequent words, we used a simple subsampling approach: each word w_i in the training set is discarded with probability computed by the formula

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}} \quad (5)$$

7

To maximize the accuracy on the phrase analogy task, we increased the amount of the training data by using a dataset with about 33 billion words. We used the hierarchical softmax, dimensionality of 1000, and the entire sentence for the context. 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

7

Interestingly, we found that the Skip-gram representations exhibit another kind of linear structure that makes it possible to meaningfully combine words by an element-wise addition of their vector representations. This phenomenon is illustrated in Table 5.

Skip-gram representation

↓

Linear structure.

↓

element-wise addition

↓

meaningfully combine words.

8

In our experiments, 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