An Examination of Skip-gram Model

Zeqiang Lai
Department of Computer Science
Beijing Institute of Technology
1120161865@bit.edu.cn

Jinxuan Jin
Department of Computer Science
Beijing Institute of Technology
1120161864@bit.edu.cn

Wenzhuo Liu
Department of Computer Science
Beijing Institute of Technology
1120161868@bit.edu.cn

Tian Huan
Department of Computer Science
Beijing Institute of Technology
1120161861@bit.edu.cn

Anteng Li
Department of Computer Science
Beijing Institute of Technology
1120161866@bit.edu.cn

Xueyan Guo
Department of Computer Science
Beijing Institute of Technology
1120162336@bit.edu.cn

Abstract

Distributed representations of words in a vector space is very useful in many natural language processing tasks. In this paper, we reproduce the Skip-gram model with negative sampling and subsampling, techniques presented by Mikolov et al. [1] that proved to be able to improve the quality of word vector representations and speed up training.

In evaluation, we give an insight into the abilities of model to learn implicitly the relationships between words. We compare the results of different types of approximation methods, including negative sampling. we also evaluate the performance of learned vector representations through entity recognition task.

1 Introduction

Section 2 outlines the Skip-gram model and the extensions that used in google's model. And in Section 3, we compare the performance of different types of candidate sampling algorithms. An visualization of word embedding and some of its interesting properties as well as an application of embedding in entity recognition task are also apresented. Finally, a brief conclusion is given in Section 4.

2 A Review of the Skip-gram Model

All the content here is some kind of a repeat of Mikolov et al. 's paper.

The main intuition in the Skip-gram model is trying to learn the word vector representations that can be used to predict their context. We can formulate this idea by maximize the log likelihood of the word embedding given a dataset. More precisely, given a sequence of training words $w_1, w_2, w_3, \cdots, w_n$, our target is to maximize this expression:

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

where c is the size of window that contains the words regarded as context, T is the total number of words in the dataset, $p(w_t + j|w_t)$ is the probability of appearance of context word w_{t+j} given its center word w_t.

In the Skip-gram model, the probability p is a function of word embedding and the basic form of it is defined as follow:

$$p(w_O|w_I) = \frac{exp(v'^T_{w_O} v_{w_I})}{\sum_{w=1}^{W} exp(v'^T_w v_{w_I})}$$

where w_O, w_I is the output and input words, v_w is the vector representation of word w, and W is the size of vocabulary.

According to this probability, we are actually going to learn a vector representation that its distance to its context is the smallest among all the words in dictionary. But the problem is that if we try to train our model through this formula, we would find training process extremely long due to the great expense of calculating $\sum_{w=1}^{W} exp(v'^T_w v_{w_I})$.

In order to accelerate the process of training, negative sampling is introduced.

2.1 Negative sampling

Instead of considering all the words in dictionary, we randomly select a small amount of "negative" words and for a center word, we want its distance to its context is smaller than the distance to these "negative" word. This is an explanation of the objective defined as follow:

$$log\sigma(v'^T_{w_O} v_{w_I}) + \sum_{i=1}^{k} E_{w_i \sim P_n(w)}[log\sigma(-v'^T_{w_i} v_{w_I})]$$

By the first term, we encourage the smaller distance between input and output words and we, in contrast, penalize the close distance between input and negative words in the second term.

$P_n(w)$ is called noise distribution and it is free to choose. According to Mikolov et al.'s work, unigram distribution raise to the $3/4$rd power outperform other distributions. Here is the precise definition,

$$P(w_i) = \frac{f(w_i)^{3/4}}{\sum_{j=0}^{n}(f(w_j)^{3/4})}$$

where$f(w)$ is the frequency of word w_i and n is the size of dictionary.

2.2 Subsampling

Words that show up too often such as "the", "of" and "for" actually provide little context to their nearby words. Thus, if we discard some of them, we can remove some of the noise from our data and in return get faster training and better representations. This process is called subsampling by Mikolov. For each word w_i in the training set, we'll discard it with probability given by

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

where t is the threshold parameter and $f(w_i)$ is the frequency of word w_i in the total dataset.

3 Experiment

3.1 Dataset

We test our models on text8[1], a cleaner and more compressive version of enwiki8[2] which only contains letters a-z and nonconsecutive spaces. For training, there are 17005207 words in total.

[1] http://mattmahoney.net/dc/text8.zip

[2] http://mattmahoney.net/dc/enwik8.zip

3.2 Training

We only train the 50000 most common words and the dimension of word embedding vectors is set to 128. The window size of context is 1 and 32 words were selected for negative sampling.

We use Gradient descent optimizer with $learning\_rate = 1$ and train the models for 100000 steps with $batch\_size = 128$ on one machine with an Intel Core i7 CPU.

3.3 Results

Table1 1 and Table2 2 shows ten closest words to "one" and "china", respectively, when different sampling method is adopted. For all the algorithms(NCE, NEG, Softmax), the results are reasonable. For example, the nearest words to "one" are the other digits and the closet words to "china" is other country or city names.

Generally, the performance of NCE(noise contrastive estimation) and Softmax(sampled softmax) are at the same level and the outcomes of the models that use subsampling are slightly better. For instance, when sampled softmax as well as subsampling are adopted, 10 closest words to "one" are the same, but the distance between "one" and "zero" is smaller than the distance between "one" and "the" when using subsampling while the result without subsampling is the opposite of the former. This is exactly what subsampling do. "The" is a very common word in English. During training, a number of "the" are discarded by subsampling and in return get better vector representations of words - "one" is closer to "zero" than "one" is to "the".

The problem is that the results of negative sampling are obviously unsatisfactory. We guess that the reason for it is that our implementation in tensorflow doesn't match the definition of NEG precisely. And we need more research on it to check.

The results on the application of our word embedding to name entity recognition show that our embedding achieves a equivalent performance with GLoVe[3] word embedding.

Table 1: The 10 closest words to "one"

Sampler	Words(ascending order)
NCE	seven, six, eight, two, four, nine, five, three, it, m
NCE—Sub	seven, two, six, five, eight, four, three, nine, god, order
NEG	two, seven, p, four, five, three, j, born, t, near
NEG—Sub	two, d, de, isbn, r, john, p, named, three, b
Softmax	two, four, eight, seven, five, six, three, nine, the, zero
Softmax—Sub	two, six, seven, four, eight, five, three, nine, zero, the

Table 2: The 10 closest words to "china"

Sampler	Words(ascending order)
NCE	australia, india, europe, canada, asia, germany, africa, front, born, east
NCE—Sub	germany, canada, america, britain, france, england, africa, australia, god
NEG	island, al, located, africa, fish, john, states, led, principal, west
NEG—Sub	east, max, students, throughout, band, u, york, italian, british
Softmax	america, australia, canada, france, europe, england, city, africa, west, order
Softmax—Sub	europe, france, australia, germany, north, canada, it, america, university, london

[3] http://nlp.stanford.edu/projects/glove/

Table 3: English NER results(CoNLL-2003 test set) using different embedding.

Experiments	F1 Score
GloVe embedding(50d)	66.47
GloVe embedding(100d)	74.57
text8 embedding(128d)	73.45

3.4 Visualization of Word embedding

From the visualization of word vectors(see figure 1), we can see:

- Letters and numbers cluster together, respectively.
- The spatial location of words pair (english, london) is similar to words pair (france, paris).
- But and however are close to each other.

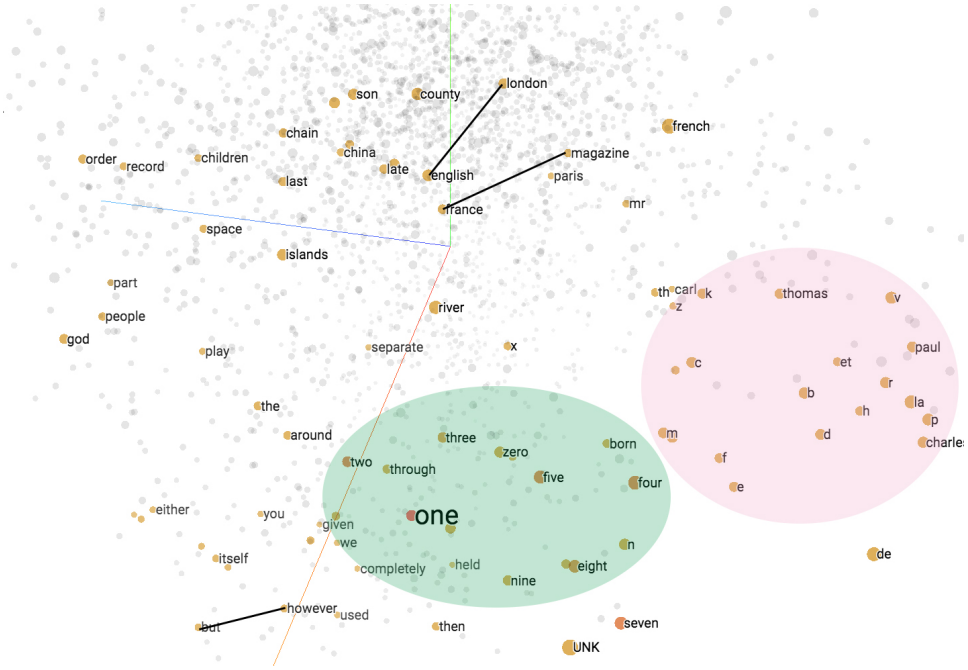

Figure 1: Word vectors projected by PCA.

The observations mentioned above indicate that the Skip-gram model is able to capture multiple different aspect of similarity between words and preserve some semantic and syntactic relationships.

4 Conclusion

In our experiment, we reproduce the results of different types of Skip-gram model. By comparison, we prove that subsampling is a good way to learn more meaningful word representations. By visualization, we show that Skip-gram model is a powerful tool to obtain word embeddings that keep the linguistic features.

References

[1] Tomas Mikolov et al. “Distributed representations of words and phrases and their compositionality”. In: *Advances in neural information processing systems*. 2013, pp. 3111–3119.