Distributed Representations of Words and Phrases and their Compositionality

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean

Ⅰ. Introduction

Word2vec 중 skip-gram의 vector representation quality를 높이고자 한다. Quality를 높이기 위해 아래와 같이 3가지 방법을 제시하였다.

1. Word 기반으로 학습된 단어를 phrase 기반으로 학습시키는 것.
2. 빈도가 높은 단어에 대한 subsampling
3. Training에 negative sampling 기법을 도입.

기존의 word 기반 학습의 문제점은 'Boston Globe'처럼 두 단어로 이루어져 있고 의미가 두 단어와 전혀 관련이 없는 phase에 대해 전혀 해석을 할 수 없다는 것이었다. 이런 phase를 idiomic하다고 하는데 이 문제점을 해결하기 위해 phase 기반의 학습을 도입하게 된다. 많은 양의 phase를 데이터 기반으로 training하고 word 대신 각 phase를 독립된 token으로 이용하여 학습시킨다. Phase를 찾기 위해 빈번하게 나타나지만 각각은 잘 나타나지 않는 단어의 score를 계산하여 찾는다. score값이 특정 threshold를 넘으면 이 phase를 독립된 token으로 여긴다.

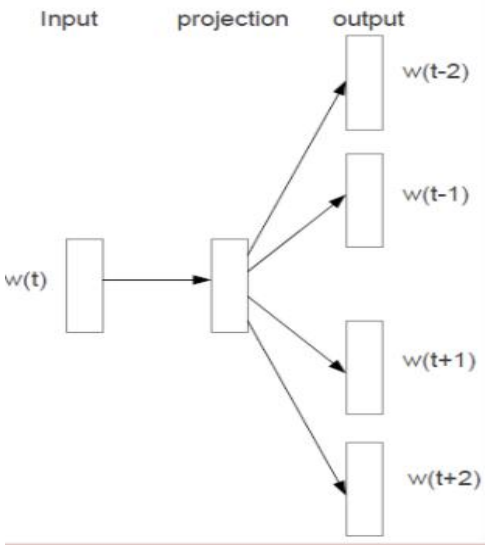
$$\text{score}(w_i, w_j) = \frac{\text{count}(w_i w_j) - \delta}{\text{count}(w_i) \times \text{count}(w_j)}$$

Threshold 값은 시행을 통해서 점점 감소시켜야 하는데 그 이유는 3개 이상의 단어로 구성된 phase를 찾기 위해서이다. (Montreal Canadiens)-(Montreal)+(Boston)=(Boston Bruins). 이전 논문처럼 phase도 analogy하게 표현이 가능하고 정확도는 72%까지 달성했다고 한다. 아래 표는 phase를 추출하도록 token을 설정한 것이다.

Czech + currency	Vietnam + capital	German + airlines	Russian + river	French + actress
koruna	Hanoi	airline Lufthansa	Moscow	Juliette Binoche
Check crown	Ho Chi Minh City	carrier Lufthansa	Volga River	Vanessa Paradis
Polish zolty	Viet Nam	flag carrier Lufthansa	upriver	Charlotte Gainsbourg
CTK	Vietnamese	Lufthansa	Russia	Cecile De

Table 5: Vector compositionality using element-wise addition. Four closest tokens to the sum of two vectors are shown, using the best Skip-gram model.

Ⅱ. The Skip-gram Model



$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t) \qquad p(w_O|w_I)=\frac{\exp\left({v'_{w_O}}^{\top}v_{w_I}\right)}{\sum_{w=1}^{W}\exp\left({v'_w}^{\top}v_{w_I}\right)}$$

Skip-gram은 한 단어로 되어있는 input을 받아 이 input 주변에 나타날 가능성이 높은 단어를 출력한다. 식에서 log probability를 높여야 정확도가 높아진다. 그러나 skip-gram의 log p를 미분하게 되면 W의 개수만큼 연산을 하게 되는데 식이 너무 길어져서 비현실적이고 시간이 굉장히 많이 소요된다.

1. Hierarchical Softmax

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

대안으로 hierarchical softmax가 있다. Sigmoid function은 0~1의 값을 가지므로 $P(W|W_i)$의 값도 0~1의 값을 갖는다. Basic skip-gram 모델과 달리 $\log P(W_0|W_i)$를 미분하면 L(W_0) 만큼의 식이 나온다. L(W_0)는 $\ln W$를 넘지 않으므로 훈련 속도가 증가한다.

2. Negative Sampling

이 논문은 hierarchical softmax가 아닌 negative sampling을 도입한다. Negative sampling은 NCE(Noise contrastive estimation)을 단순화한 버전이다. NCE와 NEG의 차이는 numerical probability가 필요 유무로 필요하지 않은 것은 특정한 값을 사용해야 한다는 제약이 없다는 의미이다.

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

NCE를 간단하게 이용해서 NEG는 NCE에 비해 정확도가 떨어진다. 하지만 이 논문은 log probability를 높여 정확도를 높이는 것이 목적이 아니고 vector representation의 quality를 높이는 것이기 때문에 NEG를 사용한다. Skip-gram은 linear하지만 non-linear한 RNN의 결과를 토대로 linear이 아닌 형태를 학습해도 linear한 연산이 가능하다. Linear한 표현이 가능하다는 것은 analogy한 표현이 가능하다는 것과 같은 뜻이다. 빈도가 낮은 단어를 대상으로 실험을 한 결과 skip-gram의 결과가 월등히 좋았다.

3. Subsampling of Frequent Words

Subsampling은 모집단이 많을 때 사용하는 방법이다. 먼저 자료를 그룹으로 나누고 각 그룹에서 표본을 추출한다. 이 과정을 통해 추출한 표본만 이용하게 되므로 시간이 절약된다. Subsampling을 사용하면 정확도가 떨어지지 않을까하는 염려가 들 수 있는데 in, the, a는 어디든지 빈번하게 등장하는 단어이지만 다른 단어에 비해서 정보와 의미를 담고 있다고 보기 어렵다. 그래서 이런 단어에 대해서는 subsampling을 해도 결과의 정확도에 크게 영향을 주지 않고 시간만 단축된다. 아래가 특정 단어가 빈번한지 아닌지 판단하는 기준이 되는 식이다.

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

f(W_i)는 frequency가 유사한 값으로 W_i라는 단어가 빈번히 나타날수록 큰 값을 가진다. 그래서 W_i가 빈번하게 나타나면 P(W_i) 값이 커지고 이 값이 t라는 threshold를 넘으면 W_i라는 word엔 subsampling을 한다. Subsampling을 한 결과를 보면 accuracy는 subsampling을 안 했을 때와 비슷하지만 소요 시간은 매우 많이 줄어든 것을 볼 수 있다. Phase 기반으로 subsampling을 했더니 정확도가 오히려 증가하였다.

https://arxiv.org/pdf/1310.4546.pdf