**Efficient Estimation of Word Representations in Vector Space**

발제자: 안유선

목차

**01** 목적

**02** Computational

Complexity

**03** Vector Accuracy

Part 1. 논문의목적

• 단어의 의미(semantic)와 문법적(syntactic) 요소를 가장 잘 보존해주는 word vector를 만드는 모

델을 찾아보자

• “What is the word that is similar to small in the same sense as biggest is similar to big?”

• Maximize accuracy, Minimize computational complexity

• 모델 종류

1. NNLM

2. RNNLM

3. CBOW

4. Skip-gram

Part 1. 논문의목적

• 좋은 vector representation이란?

• 어떤 단어를 N차원 공간 상의 벡터로

나타내었을 때, 이 공간 상에서 서로 의

미가 유사하거나 구문적 특성이 유사한

어휘들은 서로 가깝게 위치하고, 그렇지

않은 경우에는 멀리 떨어져 있는 것

Part 2. Computational Complexity

• O = Training Complexity

• E = number of training epochs

• T = number of words in the training set

• Q = Computational Complexity. 우리가 지금부터 모델별로 계산할 것

• 모델 학습 시 stochastic gradient descent 와 backpropagation 적용

Part 2. Feedforward Neural Net Language Model(NNLM)

• N = 윈도우 크기

• D = 인코딩 벡터 크기

• N \* D = projection layer 크기

• H = hidden layer 크기

• V = size of vocabulary

= output layer 크기

Part 2. Hierarchical softmax

• Hierarchical softmax를 사용하면 H \* V 를

H \* log2(V)로 줄일 수 있다

• 일반적으로 N = 10, projection layer size=

500~2000, H = 500~1000이므로 Q는

N \* D \* H의 크기에 의해 좌우된다고 봐도

무방하다

Part 2. Recurrent Neural Net Language Model (RNNLM)

• Projection layer 없이 input, hidden, output layer로만 구성

• Hidden layer의 output이 다시 다음 time step의 hidden layer의 input

으로 입력된다

• 마찬가지로 hierarchical softmax 를 사용하면 H \* V 를 H \* log2(V)로

줄일 수 있다

• 따라서 Q는 H\*H에 의해 좌우된다

Part 2. Continuous Bag-of-Words Model (CBOW)

• Input 단어의 순서가 결과값에 영향을 미치지 않는다

• 구하고자 하는 단어 뒤에 오는 단어도 input으로 받

는다

• N = 윈도우 크기

• D = 인코딩 벡터 크기

• 이 논문에서는 log-linear classifier with four future

and four history words at the input 를 사용했다

Part 2. Continuous Skip-gram Model

• CBOW와 유사하지만 Current word를

input으로 받고, 그 전후로 올 단어를

output으로 도출하는 모델

Part 3. Accuracy

• “What is the word that is similar to small in the same sense as bigges

t is similar to big?”

• Vector X = vector (biggest) – vector (big) + vector (small) 을 계산한 후

vector space에서 vector X와 가장 가까운 word vector를 output으로 산

출한다.

Part 3. Accuracy

Part 3. Accuracy 평가 방법

• A list of similar word pairs 제작

• 각 pair에 관한 질문 제작

• 총 8869 semantic & 10675 syntactic questions

• Single token words 만 사용 (ex. New York 은 X)

• 정답과 정확히 일치해야지만 옳은 것으로 인정. 유의어는 인정 X

Part 3. 모델별 Accuracy

Part 3. Publicly Available word vectors과의 비교

• Training set를 Semantic-Syntactic Word Relationship 의 question set로 확장

**감사합니다**