Word2Vec

(Efficient Estimation of Word Representation in Vector Space)

김동원

목차

논문 요약 정리 노트

논문 개요

- neural network를 통해 단어의 의미적 관계를 파악하여 숫자로 나타내는 방법에 관한 논문

- 기존의 방법을 개선하여 **단어의 양**, 단어 벡터의 **품질**, **학습 복잡도**를 개선한 두 가지 방법을 소개(CBOW, Skip-gram)

- 연산을 통해 단어 벡터의 품질을 측정하기 위한 평가 데이터 셋 정의

NNLM, RNNLM (Feedforward Neural Net Language Model, Recurrent Neural Net Language Model)

통계적 방법

- LSA : 단어-문서 행렬을 특이값 분해를 통해 단어의 잠재적 의미를 파악하는 방법
- LDA : 문서들에서 k개의 토픽을 생성해 각 단어가 어떤 토픽에 해당하는지 분석하는 방법
- N-gram : 이전 N-1개의 단어의 빈도나 연관성을 통해 다음 단어 예측

대규모 데이터에서 연산이 복잡

Neural Network 방법

- NNLM : 이전 N개의 단어를 기반으로 다음 단어를 예측하는 것을 학습하는 Neural net방식
  - Q = N x D + N x D x H + H x V
- RNNLM : NNLM에서 RNN을 이용하여 문맥정보를 추가한 방법
  - Q = H x H + H x V

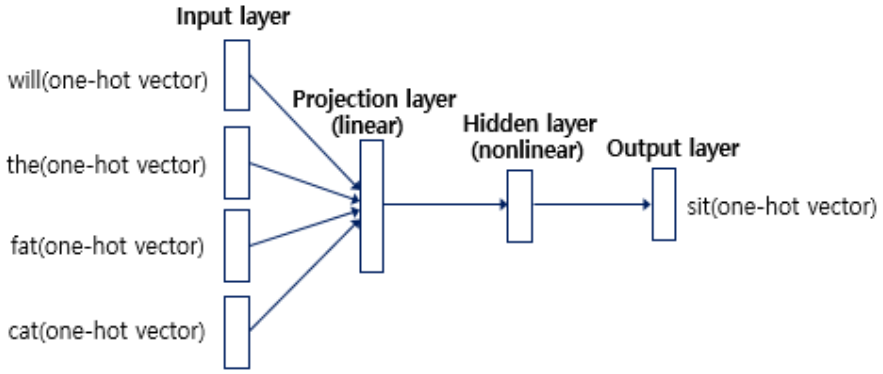
학습 연산 복잡도 O = E x T x Q (E:epochs, T:단어 수,Q:모델)

H: 단어들의 벡터 개수, D:임베딩 벡터 차원, V: 단어장 크기|원핫벡터의 크기, P: NxD크기의 차원의 projection layer, H: hidden layer크기

NNLM, RNNLM (Feedforward Neural Net Language Model, Recurrent Neural Net Language Model)

NNLM

•예문 : "what will the fat cat sit on"



x_{fat} × $W_{V\times M}$ = e_{fat}

0	0	0	1	0	0	0

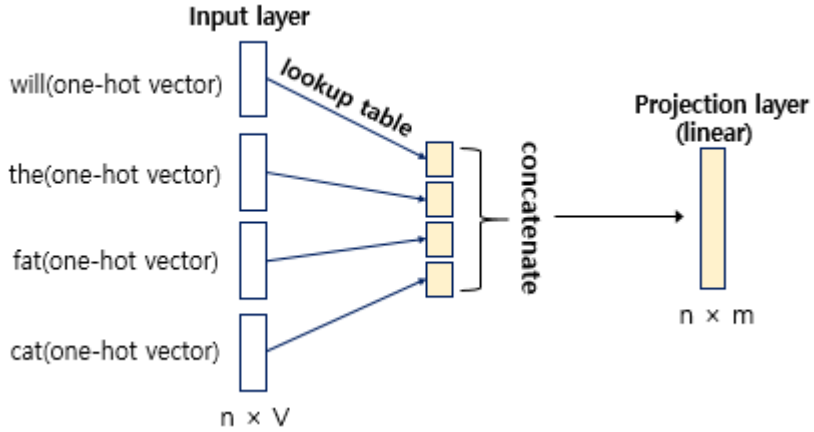
×

0.5	2.1	1.9	1.5	0.8
0.8	1.2	2.8	1.8	2.1
0.1	0.8	1.2	0.9	0.7
2.1	1.8	1.5	1.7	2.7

=

2.1	1.8	1.5	1.7	2.7

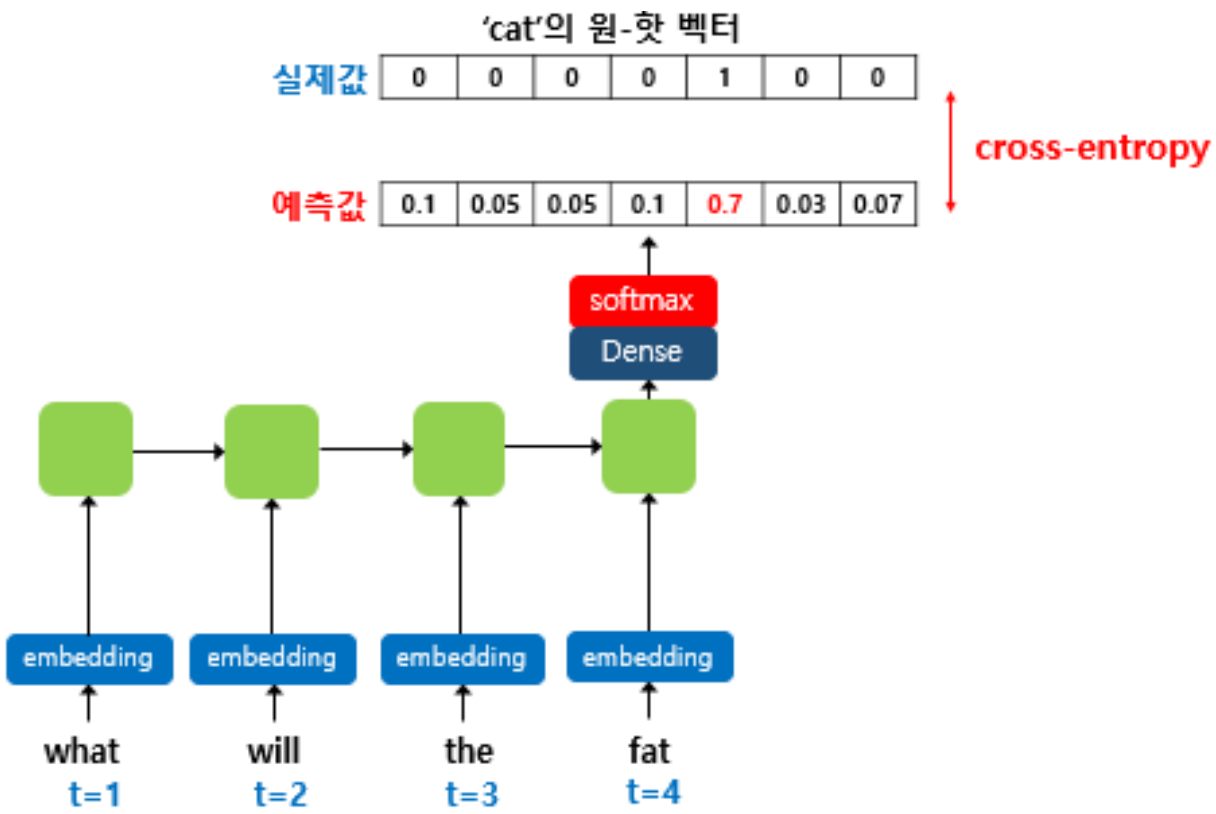
lookup table



NNLM, RNNLM (Feedforward Neural Net Language Model, Recurrent Neural Net Language Model)

RNNLM

•예문 : "what will the fat cat sit on"

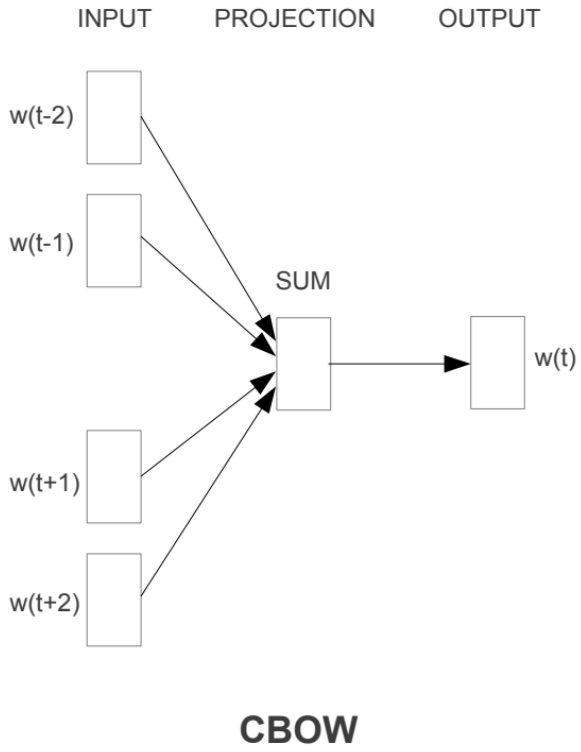


CBOW(Continuous Bag-of-Words Model), Skip-gram

논문에서 제안한 대규모 단어 데이터를 학습하기 위한 두 가지 방법

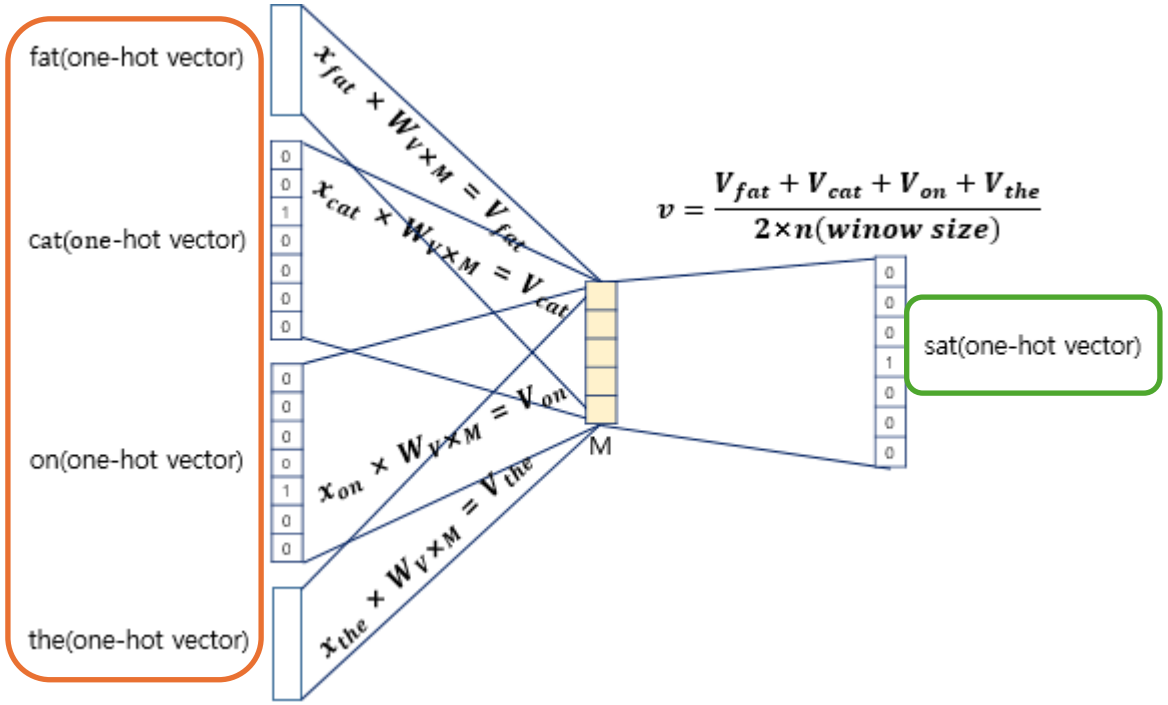
비선형 변환 layer에서의 연산이 연산 복잡도에서 큰 비중을 차지하여 이를 제거

CBOW: 주변 단어를 통해 중심 단어 예측



CBOW

•예문 : "The fat cat sat on the mat"

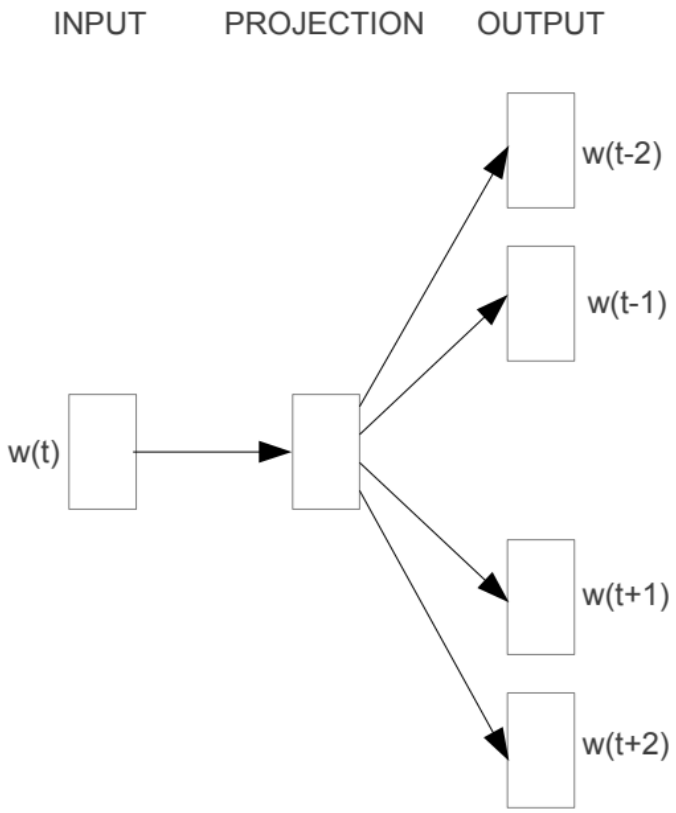


Q : N x D + D x V

CBOW(Continuous Bag-of-Words Model), Skip-gram

Skip-gram: 중심 단어를 통해 주변 단어를 예측

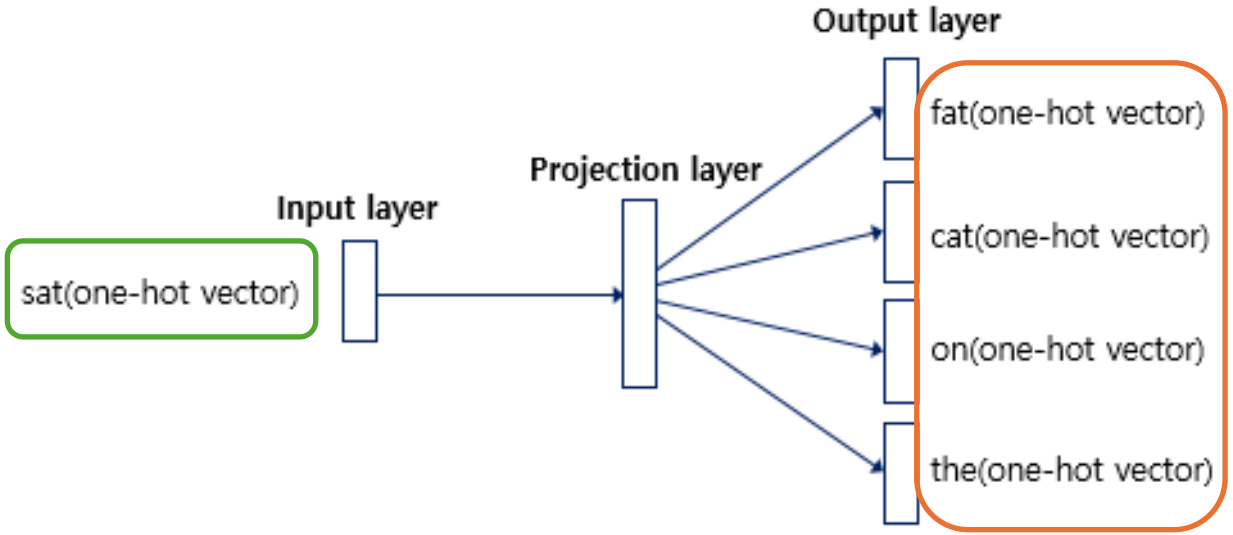
가까울 수록 가중치가 크게 학습하고 넓은 범위의 단어를 예측하게 할 수록 단어벡터의 품질이 좋아진다



Skip-gram

Q : C x (D + D x V)

•예문 : "The fat cat sat on the mat"



Hierarchical softmax

Q ..

NNLM = N x D + N x D x H + H x V

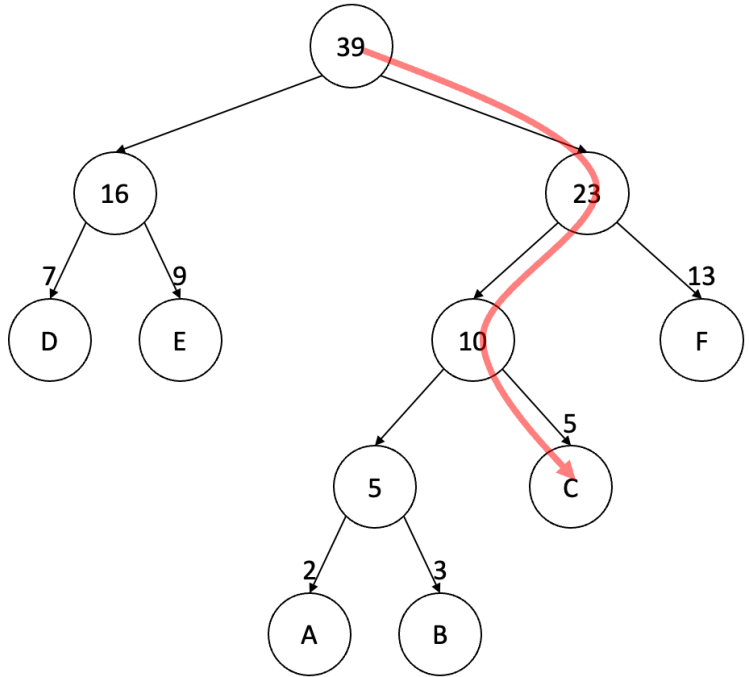
RNNLM = H x H + H x V

CBOW = N x D + D x V

Skip-gram = C x (D + D x V)

(H x V), (D x V)부분은 softmax를 통해 실제 단어들의 예측값을 생성하는데, 단어가 많을 수 록 연산이 복잡해지므로 계층적 softmax를 적용하여 연산량을 줄인다.

Huffman coding을 통해 단어들을 tree형태로 만들어 softmax를 계산하게 되면 x $\log_2(v)$로 줄일 수 있게 된다.



Result

단어 벡터는 의미를 파악하여 생성되어 연산을 통해 여러 단어들의 의미관계를 파악할 수 있다.

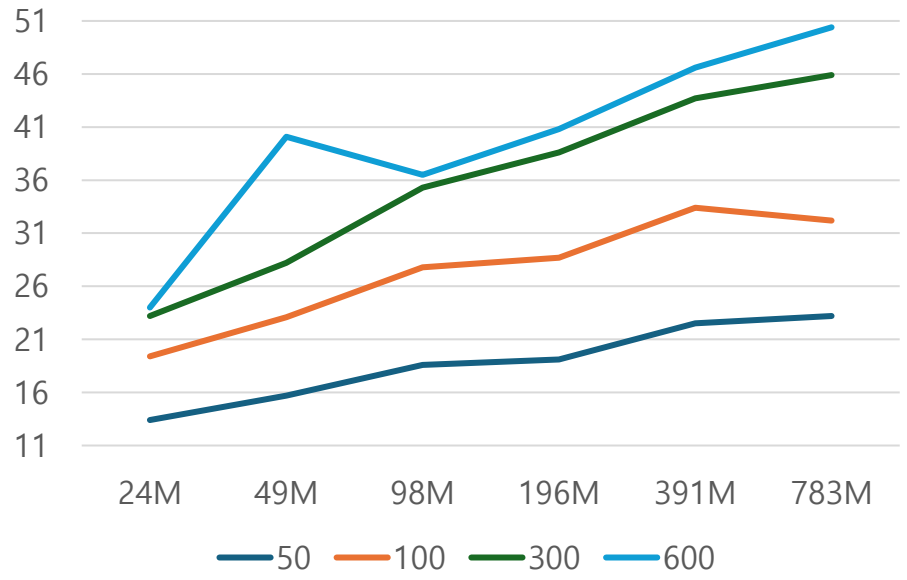
예시) vector("왕") – vector("남자") + vector("여자") = vector("여왕")

의미적, 문법적 관계를 가지는 두쌍을 통해 평가 데이터셋을 정의 하여 평가하였다.

Type of relationship	Word Pair 1		Word Pair 2	
Common capital city	Athens	Greece	Oslo	Norway
All capital cities	Astana	Kazakhstan	Harare	Zimbabwe
Currency	Angola	kwanza	Iran	rial
City-in-state	Chicago	Illinois	Stockton	California
Man-Woman	brother	sister	grandson	granddaughter
Adjective to adverb	apparent	apparently	rapid	rapidly
Opposite	possibly	impossibly	ethical	unethical
Comparative	great	greater	tough	tougher
Superlative	easy	easiest	lucky	luckiest
Present Participle	think	thinking	read	reading
Nationality adjective	Switzerland	Swiss	Cambodia	Cambodian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars
Plural verbs	work	works	speak	speaks

Result

임베딩 차원과 학습 데이터가 같이 증가할 수록 성능이 좋아진다.



Model Architecture	Semantic-Syntactic Word Relationship test set		MSR Word Relatedness Test Set [20]
	Semantic Accuracy [%]	Syntactic Accuracy [%]	
RNNLM	9	36	35
NNLM	23	53	47
CBOW	24	64	61
Skip-gram	55	59	56

Model	Vector Dimensionality	Training words	Accuracy [%]			Training time [days]
			Semantic	Syntactic	Total	
3 epoch CBOW	300	783M	15.5	53.1	36.1	1
3 epoch Skip-gram	300	783M	50.0	55.9	53.3	3
1 epoch CBOW	300	783M	13.8	49.9	33.6	0.3
1 epoch CBOW	300	1.6B	16.1	52.6	36.1	0.6
1 epoch CBOW	600	783M	15.4	53.3	36.2	0.7
1 epoch Skip-gram	300	783M	45.6	52.2	49.2	1
1 epoch Skip-gram	300	1.6B	52.2	55.1	53.8	2
1 epoch Skip-gram	600	783M	56.7	54.5	55.5	2.5

Dimensionality / Training words	24M	49M	98M	196M	391M	783M
50	13.4	15.7	18.6	19.1	22.5	23.2
100	19.4	23.1	27.8	28.7	33.4	32.2
300	23.2	29.2	35.3	38.6	43.7	45.9
600	24.0	30.1	36.5	40.8	46.6	50.4

CBOW가 Skip-gram보다 학습 빠르고 두 모델 모두 임베딩 벡터가 증가할 경우에 학습이 오래 걸림

Result

문장에서 빠진 하나의 단어를 예측하는 Microsoft Sentence Completion Challenge에서의 모델별 성능

Architecture	Accuracy [%]
4-gram [32]	39
Average LSA similarity [32]	49
Log-bilinear model [24]	54.8
RNNLMs [19]	55.4
Skip-gram	48.0
Skip-gram + RNNLMs	**58.9**

Skip-gram과 RNNLMs를 weighted combination한 것이 sota를 달성

참고자료link

LSA,LDA - heeee__ya-Tistory

NNLM, RNNLM, word2vec 이미지-(도서) 딥 러닝을 이용한 자연어 처리 입문

Word2Vec paper review–(유튜브)서울대학교 산업공학과 DSBA 연구실