Doc2vec

발표자: 박채은(DSL 5기)

Doc2vec이란

Doc2vec :

Document embedding with paragraph vectors

document 임베딩 모델


paragraph1


paragraph2


paragraph3


paragraph4

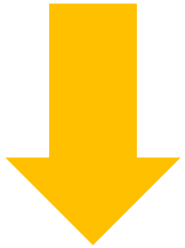
…


paragraphN

목표: Paragraph vector 구하기

Doc2vec: w2v을 확장해서 만든 모델

Word2vec
: 단어 임베딩 모델
훈련 방식: CBOW, Skip-gram



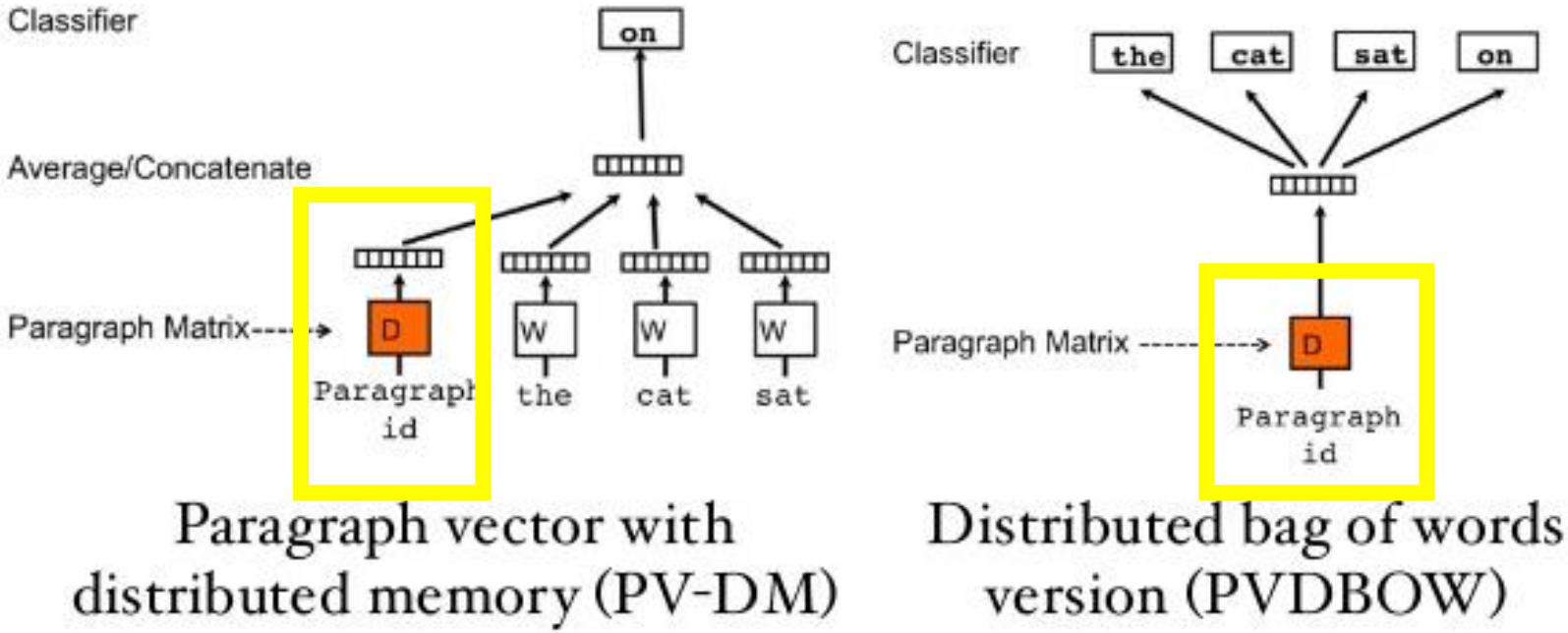
확장

Doc2vec
Word2vec의 장점을 그대로 가지게 됨.
훈련방식이 유사

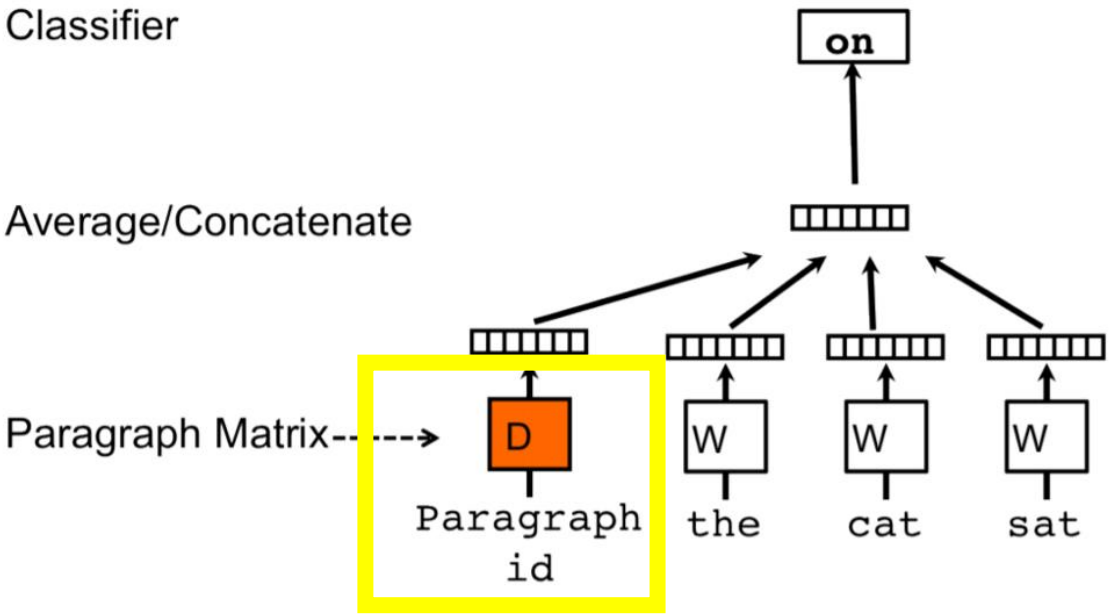
Doc2vec: w2v을 확장해서 만든 모델

W2v model + 각각의 문서를 나타내는 paragraph_id
-> paragraph vector 얻을 수 있음.

훈련 방식: PV-DM, PV-DBOW

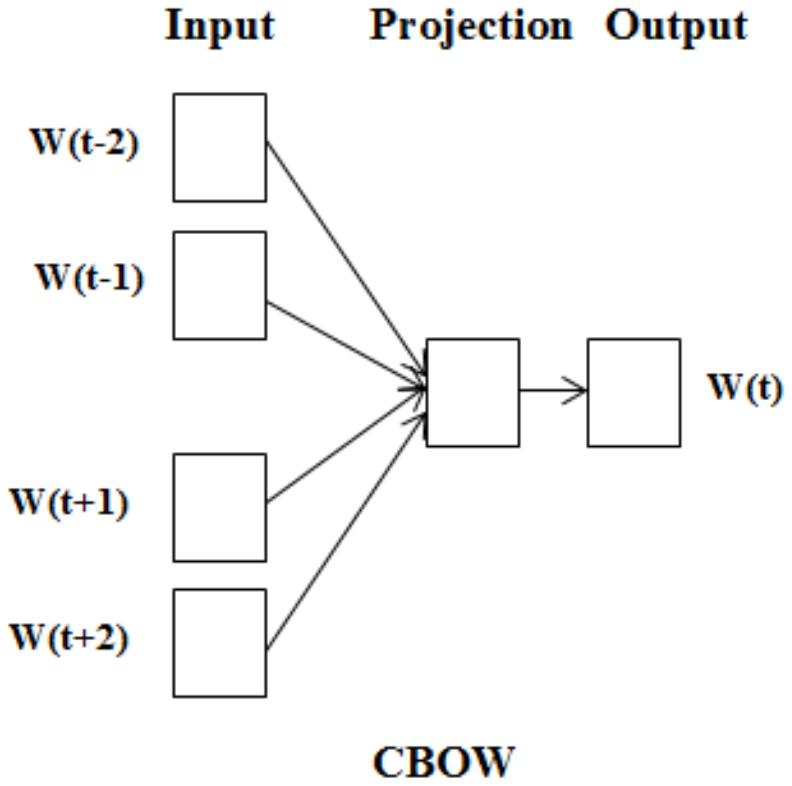


PV-DM
(paragraph vector with distributed memory)



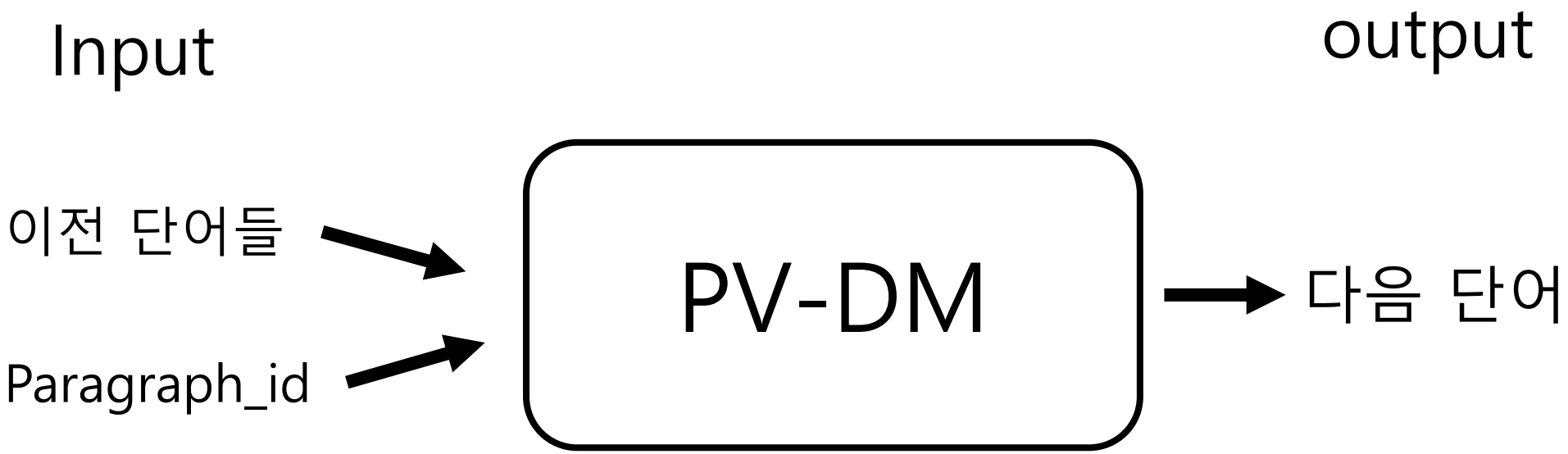
d2v

VS



w2v

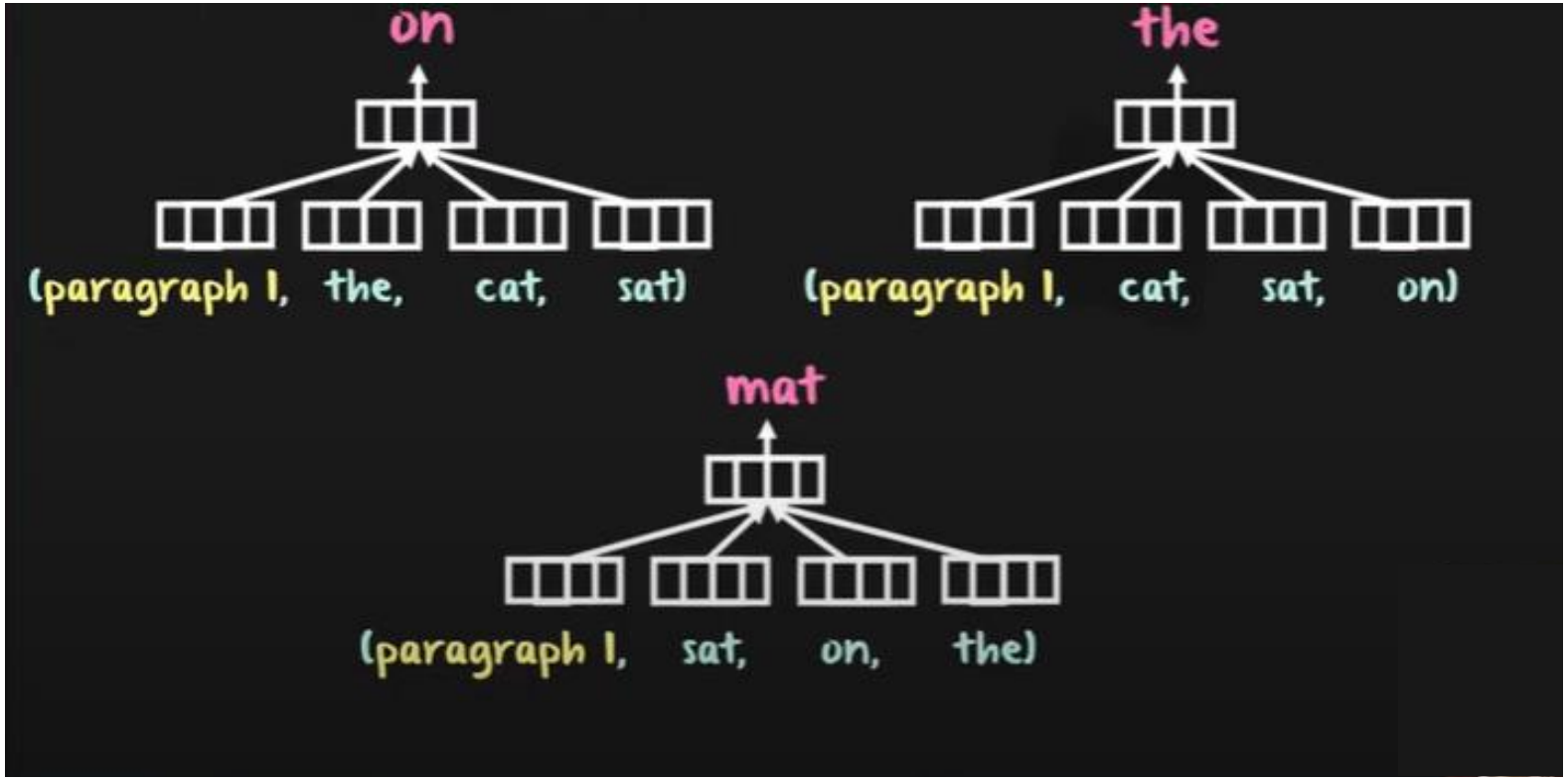
PV-DM
(paragraph vector with distributed memory)



PV-DM
(paragraph vector with distributed memory)

- 예) Paragraph_1 : ‘the cat sat on the mat.’

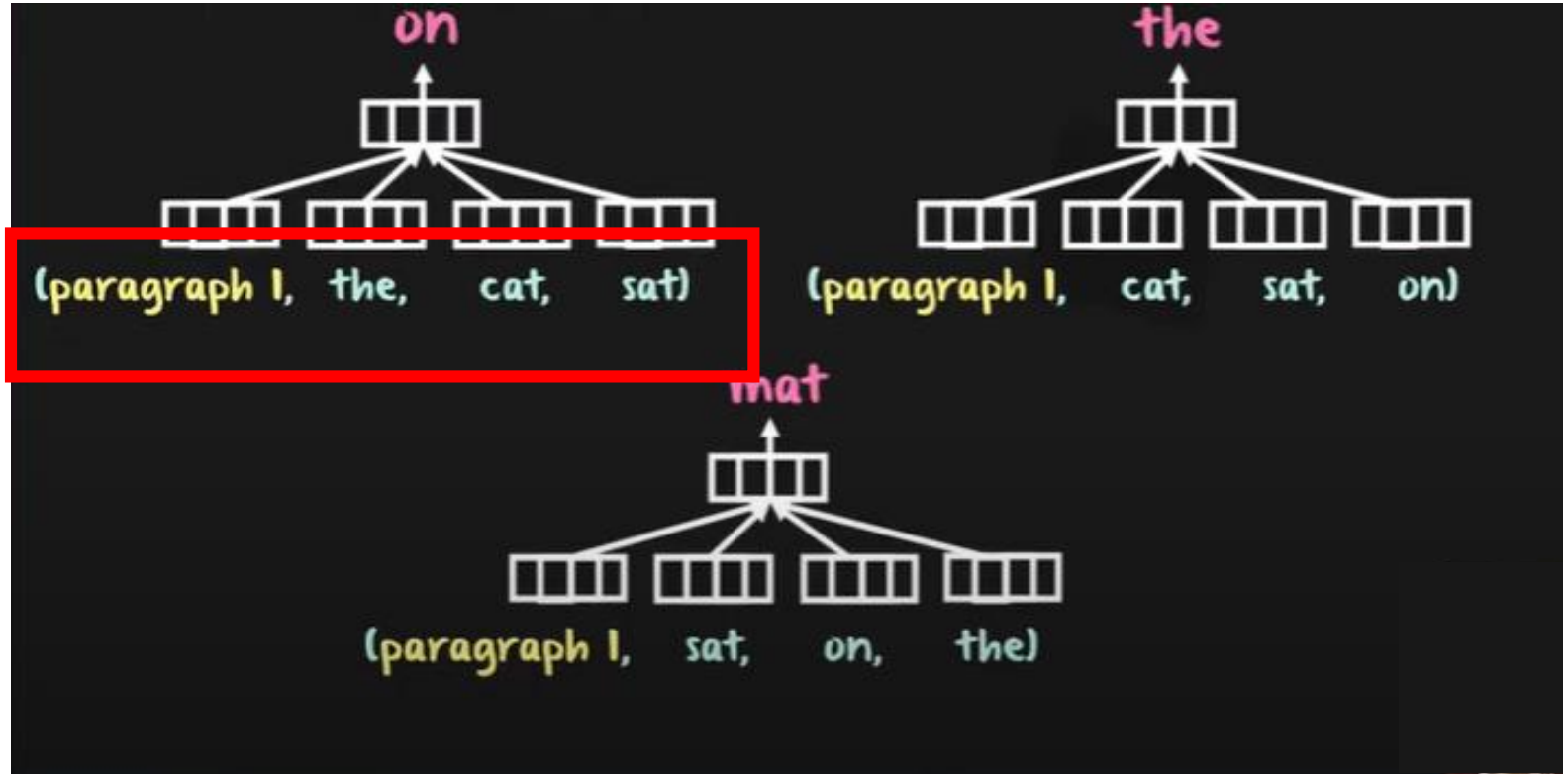
  윈도우 사이즈: 3



PV-DM
(paragraph vector with distributed memory)

- 예) Paragraph_1 : 'the cat sat on the mat.'

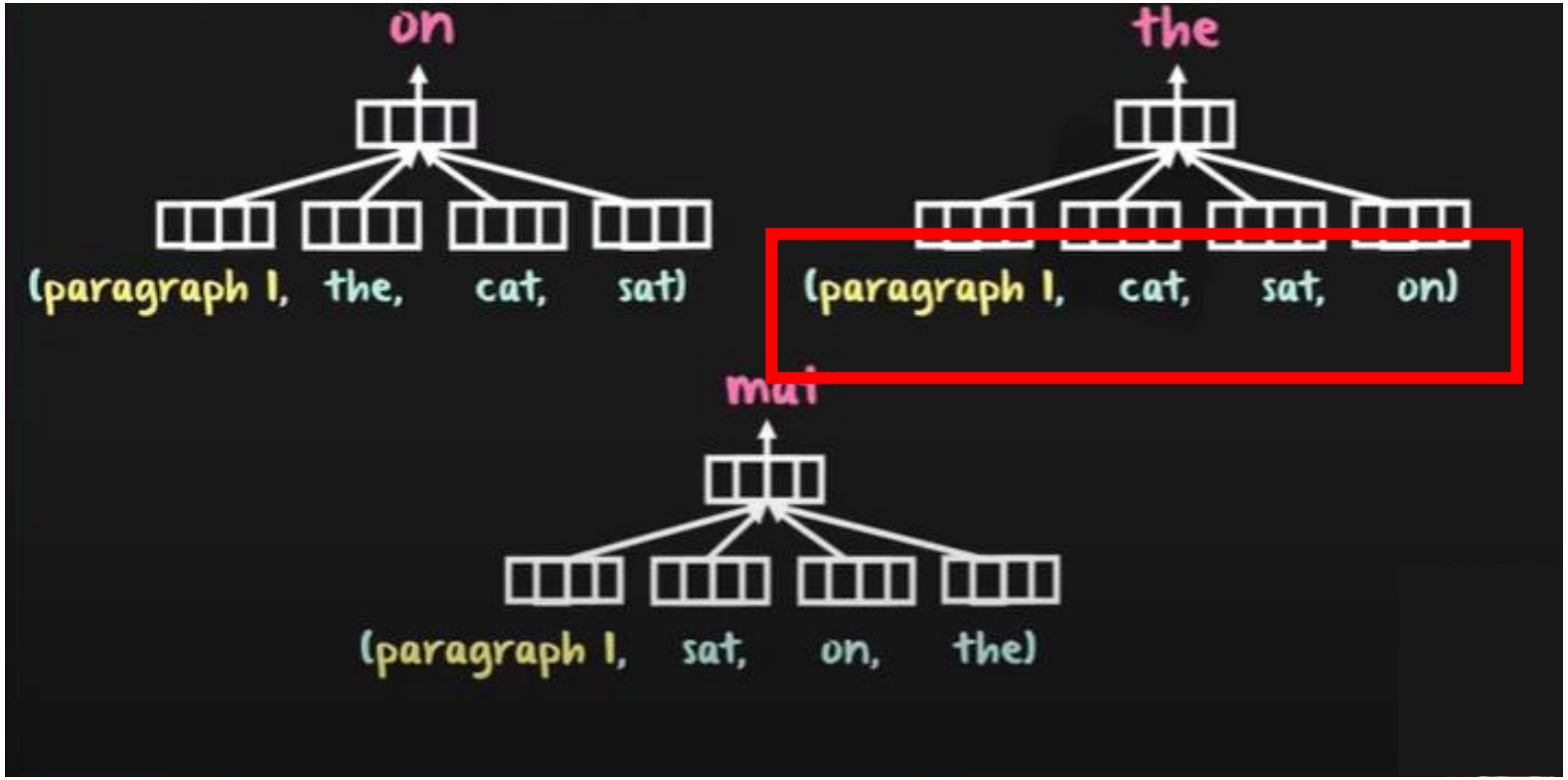
  윈도우 사이즈: 3



PV-DM
(paragraph vector with distributed memory)

- 예) Paragraph_1 : ‘the cat sat on the mat.’

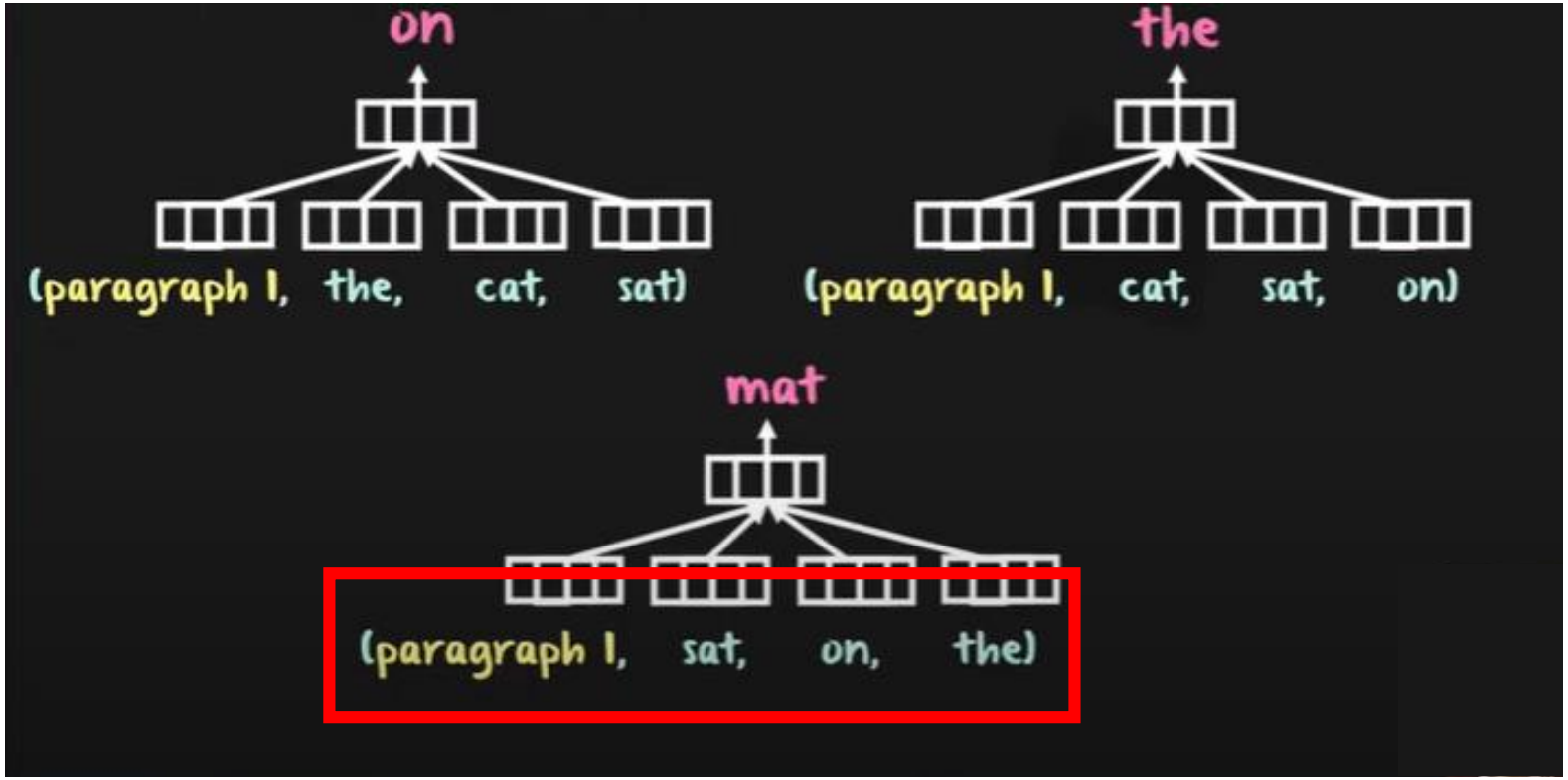
  윈도우 사이즈: 3



PV-DM
(paragraph vector with distributed memory)

- 예) Paragraph_1 : ‘the cat sat on the mat.’

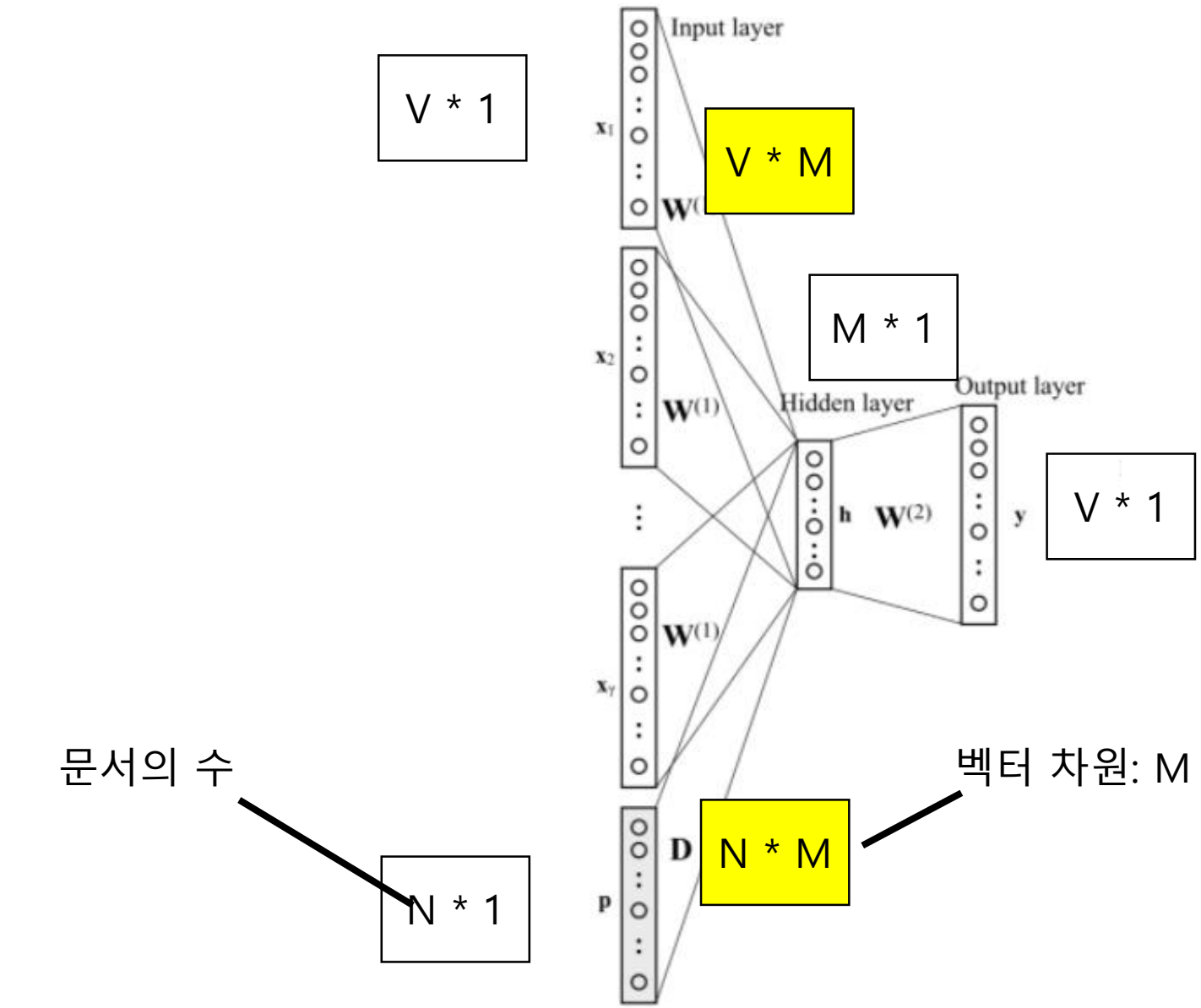
  윈도우 사이즈: 3



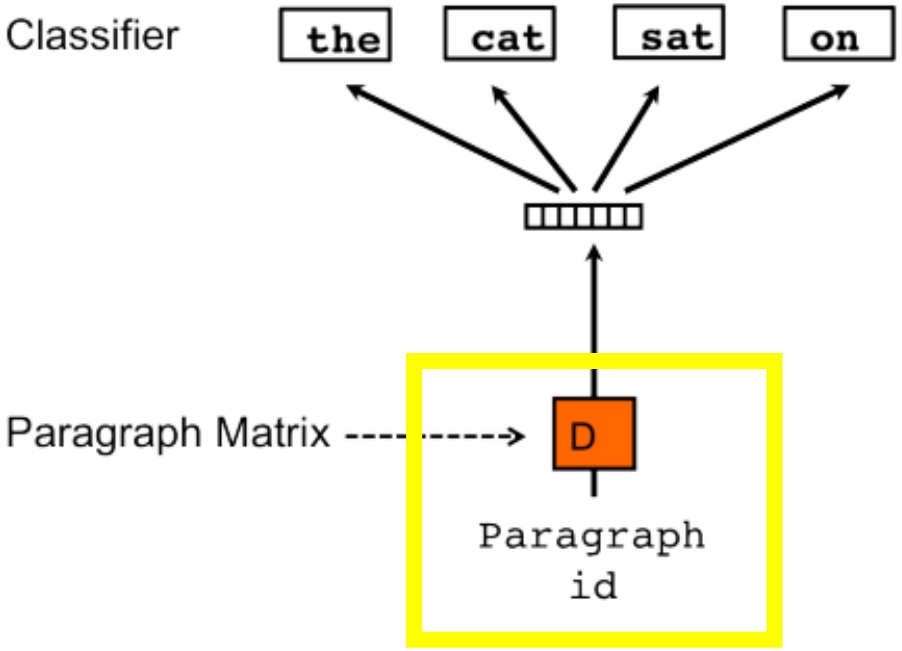
PV-DM
(paragraph vector with distributed memory)

PV-DM이라는 이름이 붙은 이유는?

PV_DM을 통해 만들어진 벡터가
문서의 주제를 잡아주는 저장소의 역할을 하기 때문에

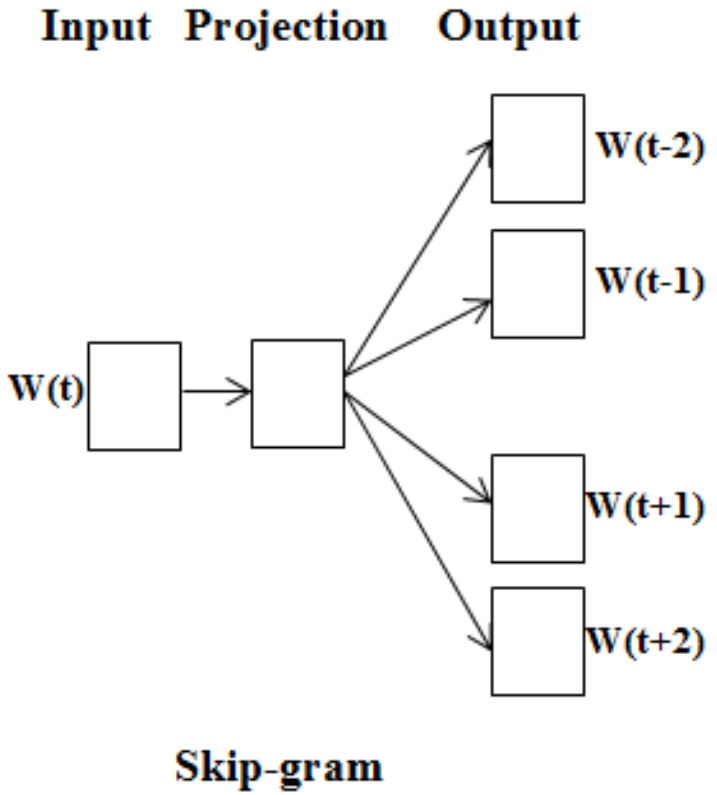

V * 1
Input layer
V * M
M * 1
Hidden layer
Output layer
V * 1
문서의 수
N * 1
N * M
벡터 차원: M

PV-DBOW
(Paragraph Vector with Distributed Bag Of Words)



d2v

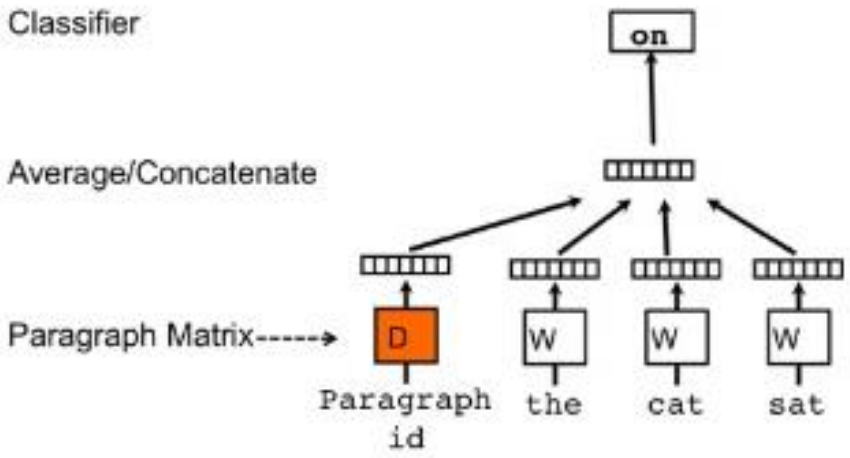
vs



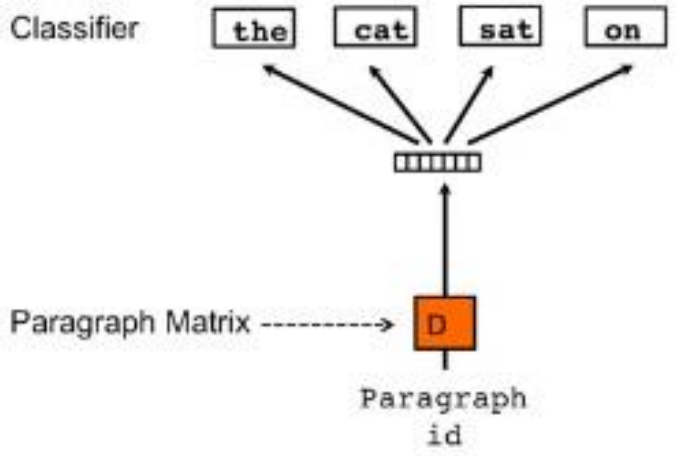
w2v

PV-DM과 PV-DBOW의 차이점

- word vector 생성 여부



Paragraph vector with distributed memory (PV-DM)



Distributed bag of words version (PVDBOW)


Thank
You!