Embedding Summary

서민지(jaaaamj0711) 유태혁(yth01)

NPLM(Leural Probabilistic Language Model)

단어 시퀀스가 주어졌을 때 직전까지 등장한 n-1개 단어들로 다음 단어를 맞추는 n-gram 언어 모델

확률 벡터에서 가장 높은 요소의 인덱스에 해당하는 단어가 실제 정답 단어와 일치하도록 역전파를 통해 학습한다.

↑

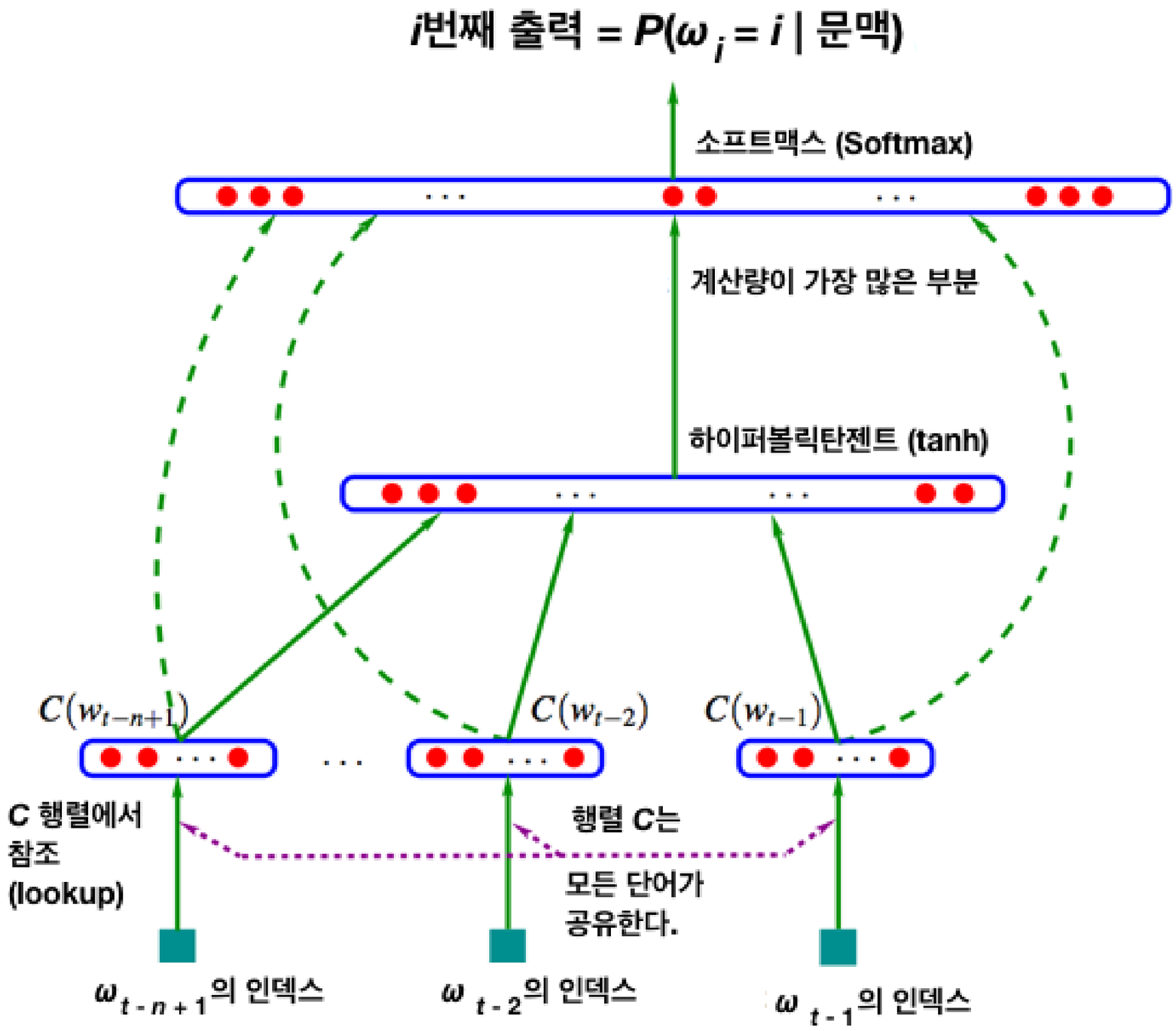
소프트맥스 함수를 적용하여 확률 벡터로 바꿔준다

↑

활성화 함수 tanh를 사용하여 데이터를 비선형 구조로 바꿔준다.

↑

w_t는 t번째 단어에 대한 One-hot-Vector이다. 각각의 단어에 해당하는 인덱스값을 불러온다. 인덱스에 해당하는 행렬C의 열벡터를 불러온다. 각각의 단어에 해당하는 열벡터를 C에서 참조해 묶어준다.



학습 파라미터가 너무 많아 계산이 복잡해지고, 자연스럽게 모델 과적합(overfitting)이 발생할 가능성이 높다는 한계를 가진다.

Word2vec

"단어의 주변을 보면 그 단어를 안다"라는 predictive method로
CBOW 또는 Skip-Gram 알고리즘으로 학습하여 예측하는 모델

CBOW: 주변에 있는 문맥 단어를 가지고 타깃단어를 예측

윈도우 크기 범위 안에 있는 주변 단어들의
로 one-hot vector로 만들어 Input에 들어간다

↓

Input에 가중치 W를 곱한다.

↓

평균을 구하여 평균 벡터를 생성한다.

↓

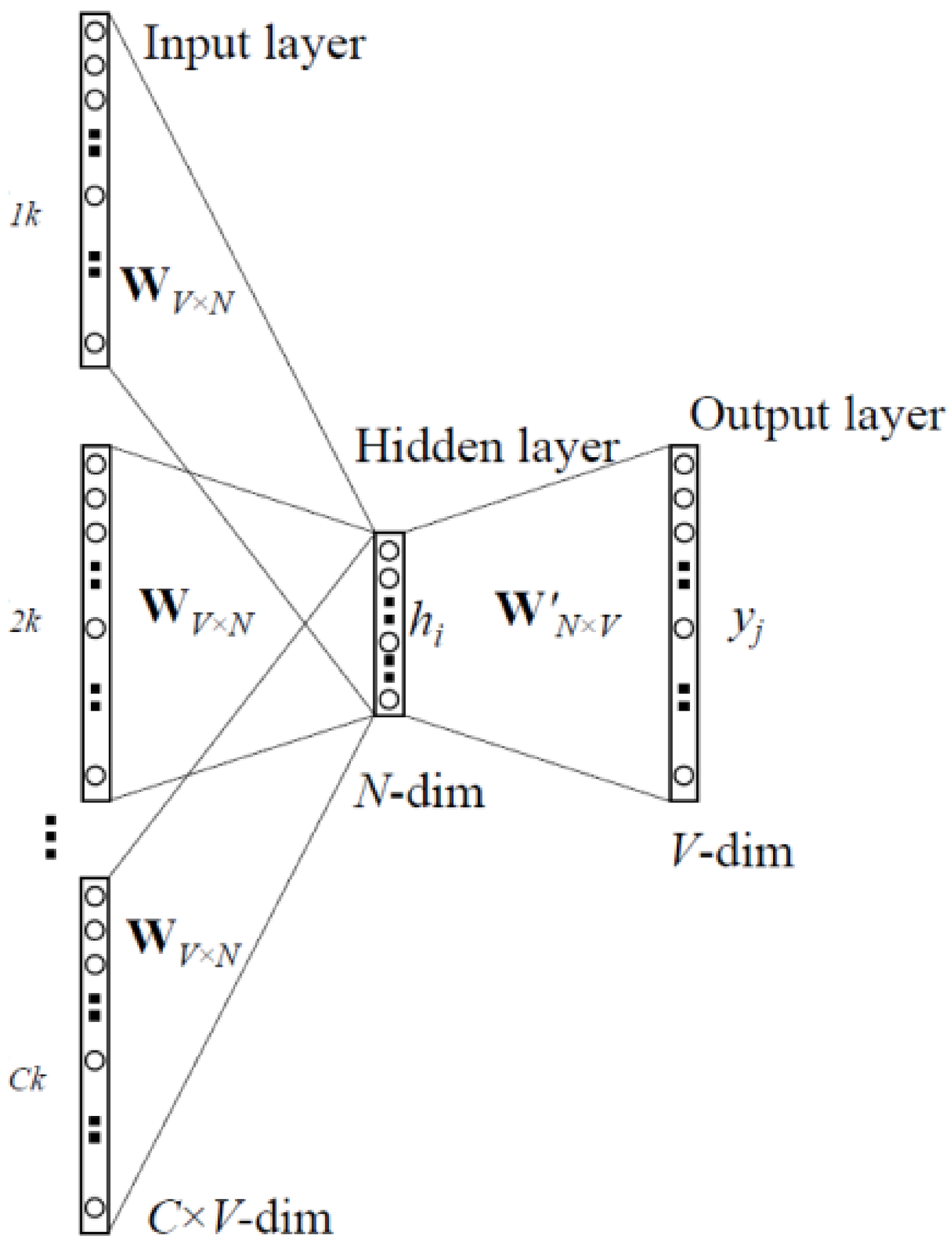
평균 벡터는 두번째 가중치 행렬인
W'와 곱해진다.

↓

확률로 표현하기 위해서 softmax 함수를 취한다.

↓

loss를 통해 단어의 one-hot vector와의
오차를 측정한다.



Word2vec

Skip-gram: 타킷 단어를 가지고 주변 문맥 단어를 예측

윈도우 크기 범위 안에 있는 주변 단어들의
로 one-hot vector로 만들어 Input에 들어간다

↓

Input에 가중치 W를 곱한다.

↓

평균 벡터는 두번째 가중치 행렬인
W'와 곱해진다.

↓

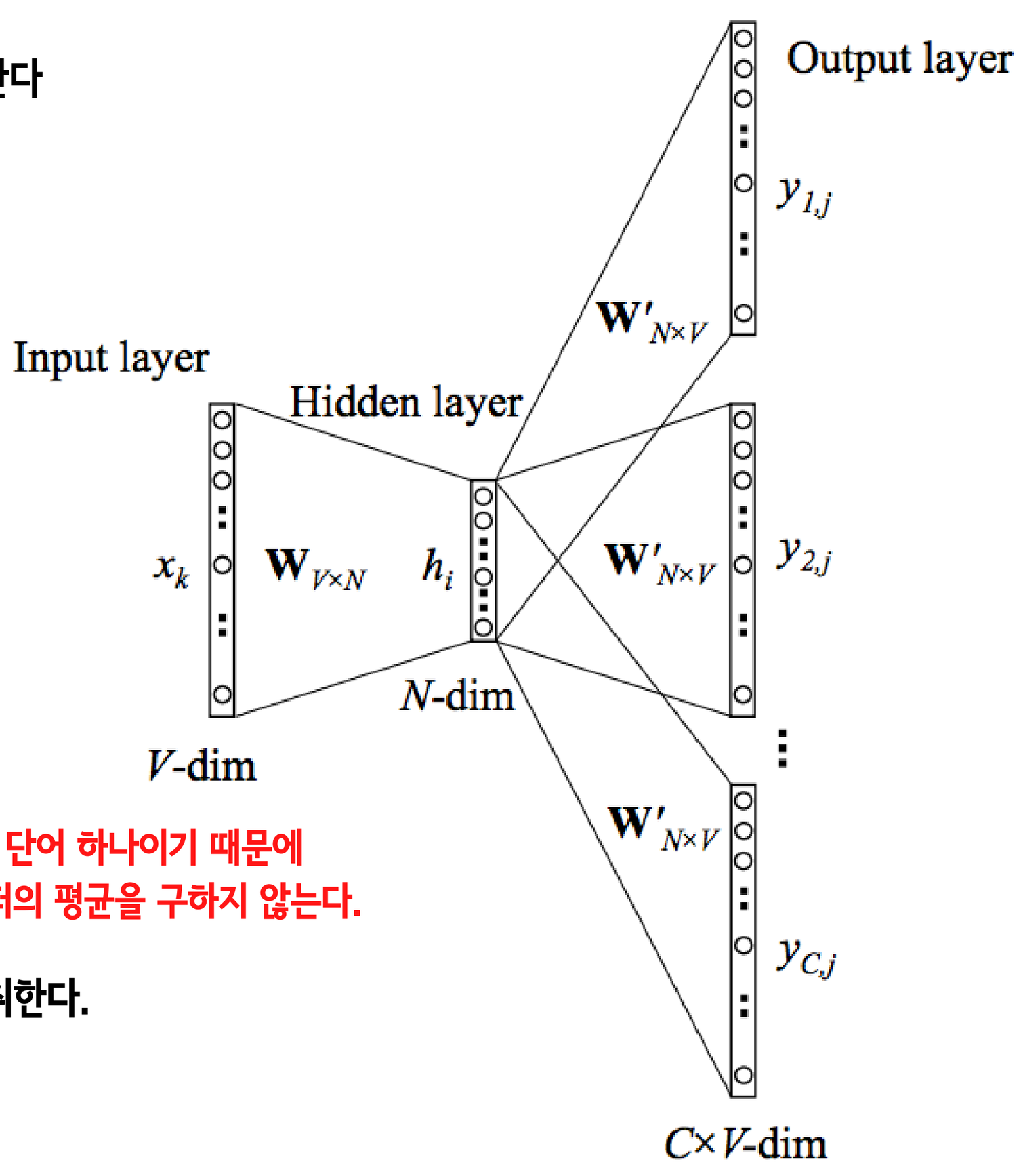
입력이 중심 단어 하나이기 때문에
은닉층에서 벡터의 평균을 구하지 않는다.

↓

확률로 표현하기 위해서 softmax 함수를 취한다.

↓

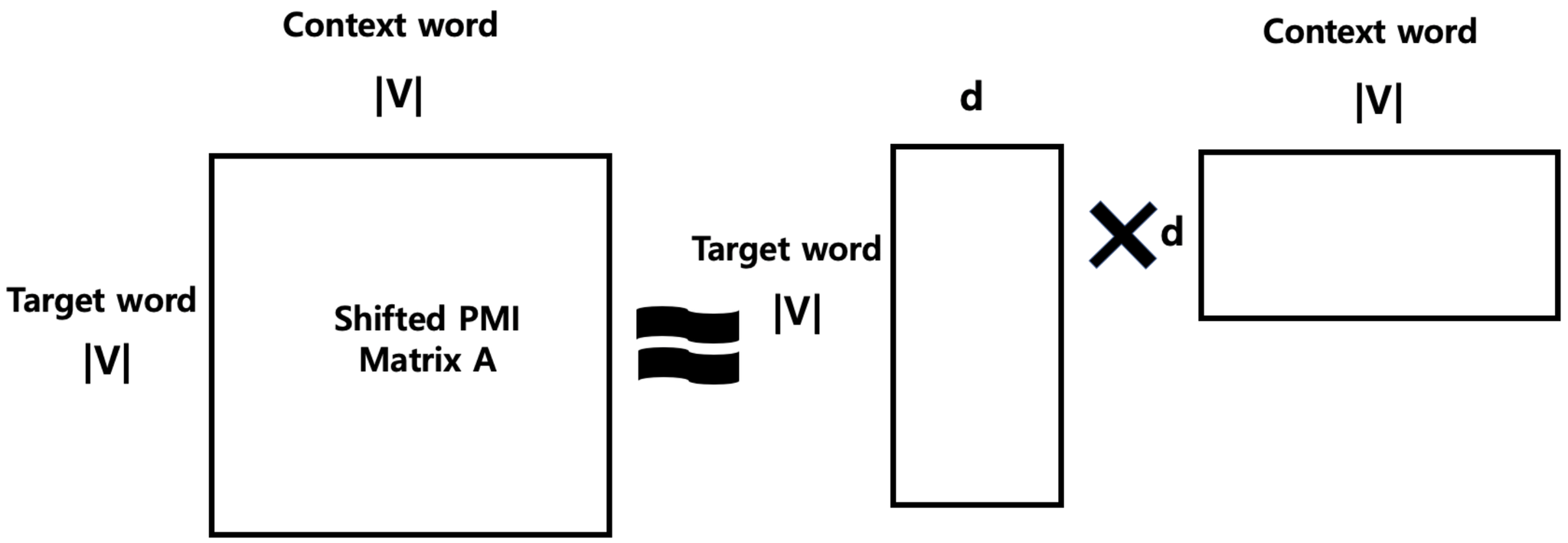
loss를 통해 단어의 one-hot vector와의
오차를 측정한다.



사용자가 지정한 윈도우에서 학습을 하기 때문에 말뭉치 전체의 의미를 파악하기 어렵고,
어휘가 많아지면 softmax함수를 사용하는데 어려움이 있다.

Word2vec

네거티브 샘플링(negative sampling) 기법으로 학습된 Word2Vec의 Skip-gram 모델은 Shifted PMI 행렬을 분해한 것과 같다는 것을 볼 수 있다.



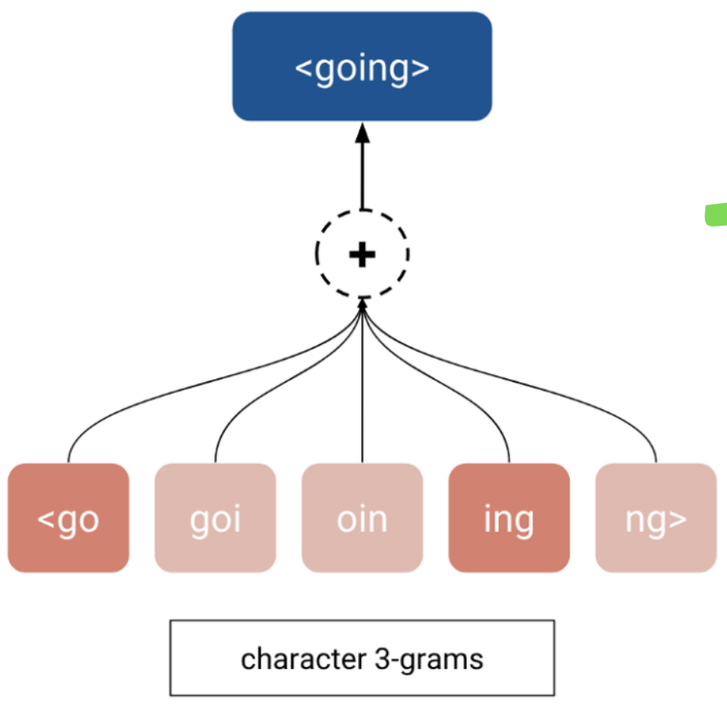
A_{ij} 는 SPMI 행렬의 i, j번째 원소이다. k는 Skip-gram 모델의 negative sample 수를 의미한다. 그러므로 k=1인 negative sample 수가 1개인 Skip-gram 모델은 PMI 행렬을 분해하는 것과 같다.

$$A_{ij}^{SGNS} = U_i \cdot V_j = PMI(i,j) - logk$$

Swivel(Submatrix-Wise

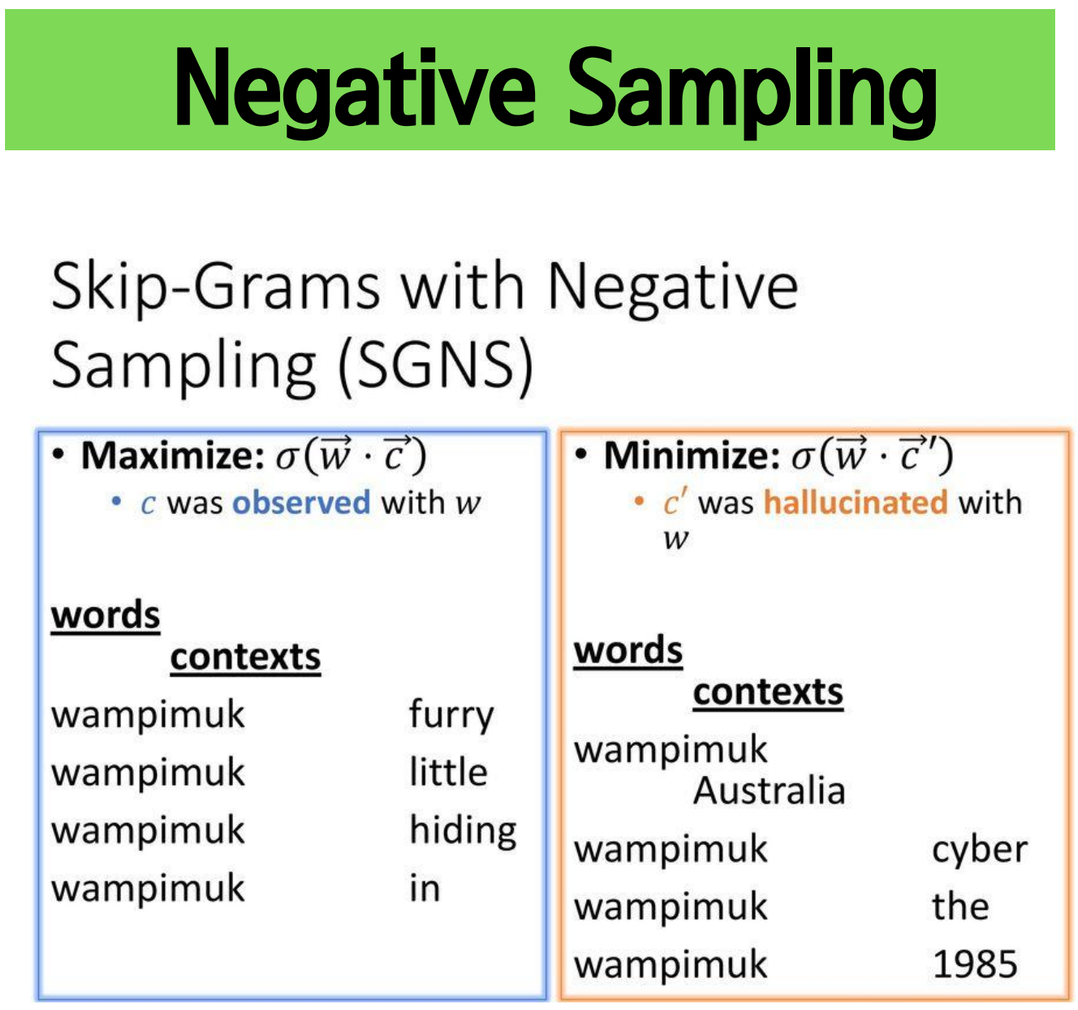
word를 subword 단위로 표현하는 것으로 기본적으로
(Skip-Gram with Negative Sampling, SGNS) 방식이다.

subword



n-gram(n=3)인 문장의 예시로
FastText 모델에서는 단어의 임베딩을
n-gram 벡터의 합으로 표현한다.

Negative Sampling



목표는 정답 문맥 단어가 등장할 확률을 높이고
나머지 단어들은 감소하는 방향으로 훈련한다.

타깃 단어와 문맥 단어의 쌍을 이진 분류
(positive, negative)하는 과정에서 학습이 된다.

softmax는 모델을
1스텝에서 전체 단어를 모두 계산한다.

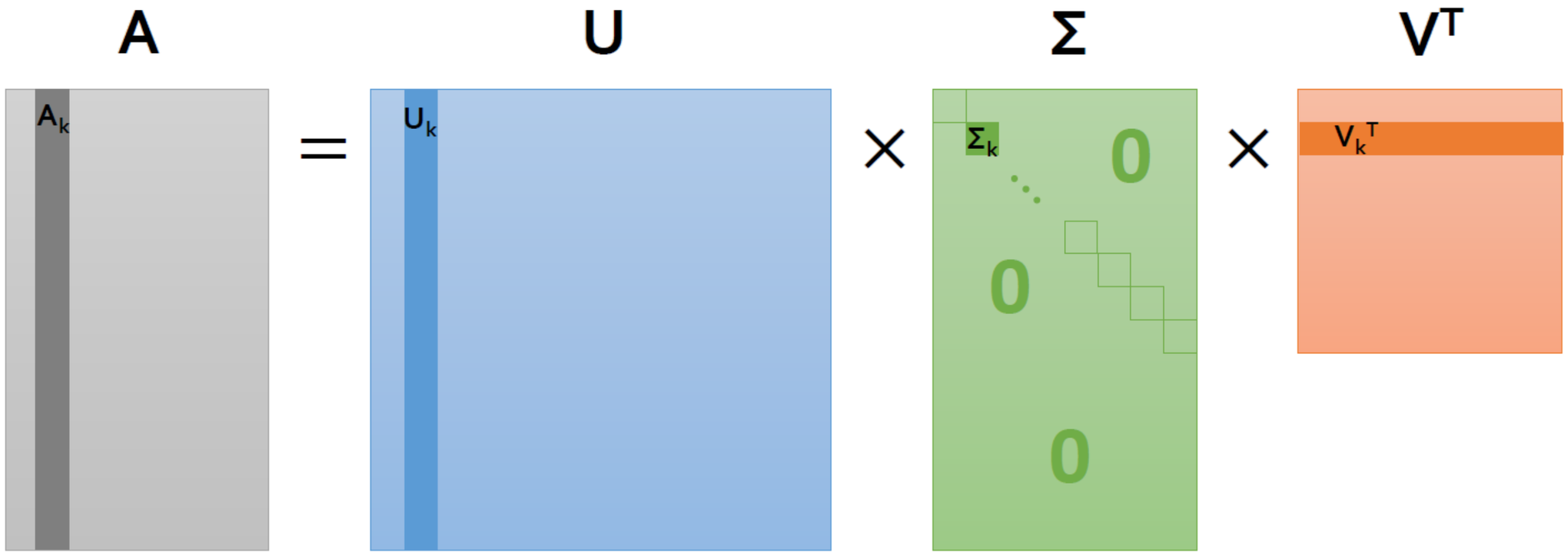
1개의 positive와 k개의 negative만 계산하게
되면서 계산량이 압도적으로 줄어든다.

FastText subword 단위로 단어를 분리하기 때문에
OOV(Out of Vocabulary)에 강한 장점이 있다.

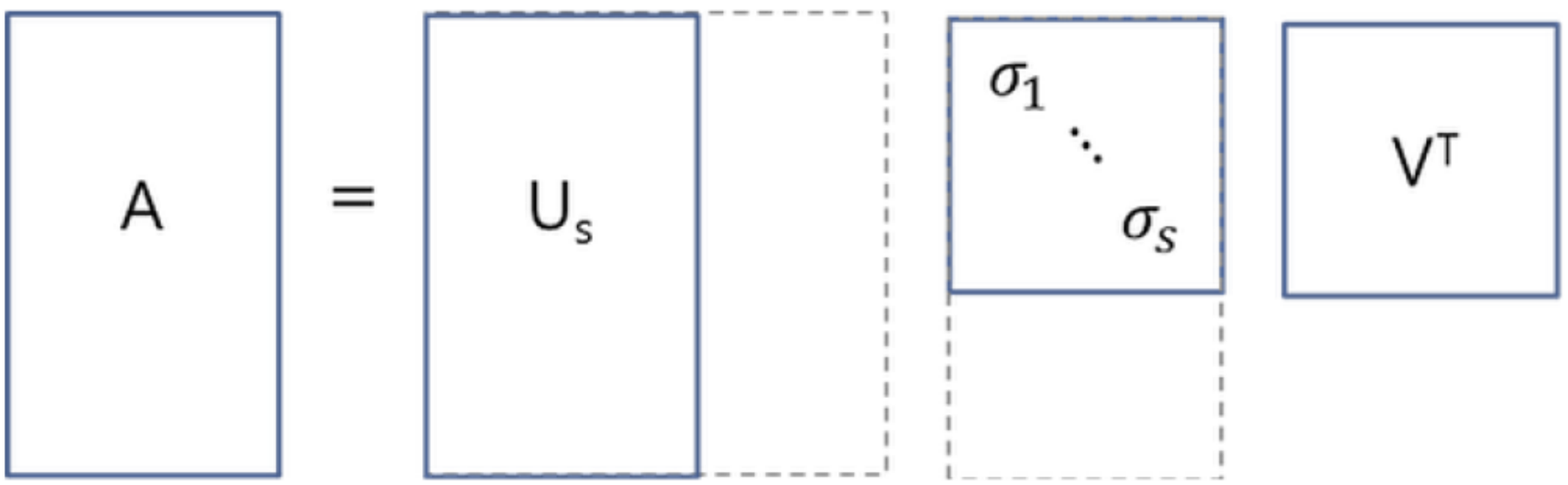
잠재 의미 분석 - LSA(Latent Semantic Analysis)

단어-문서 행렬, TF-IDF 행렬, 단어-문맥 행렬같은 커다란 행렬에 특이값 분해를 수행해 데이터의 차원축소로 계산 효율성을 키우고 잠재 의미를 이끌어내기 위한 방법론

특이값 분해

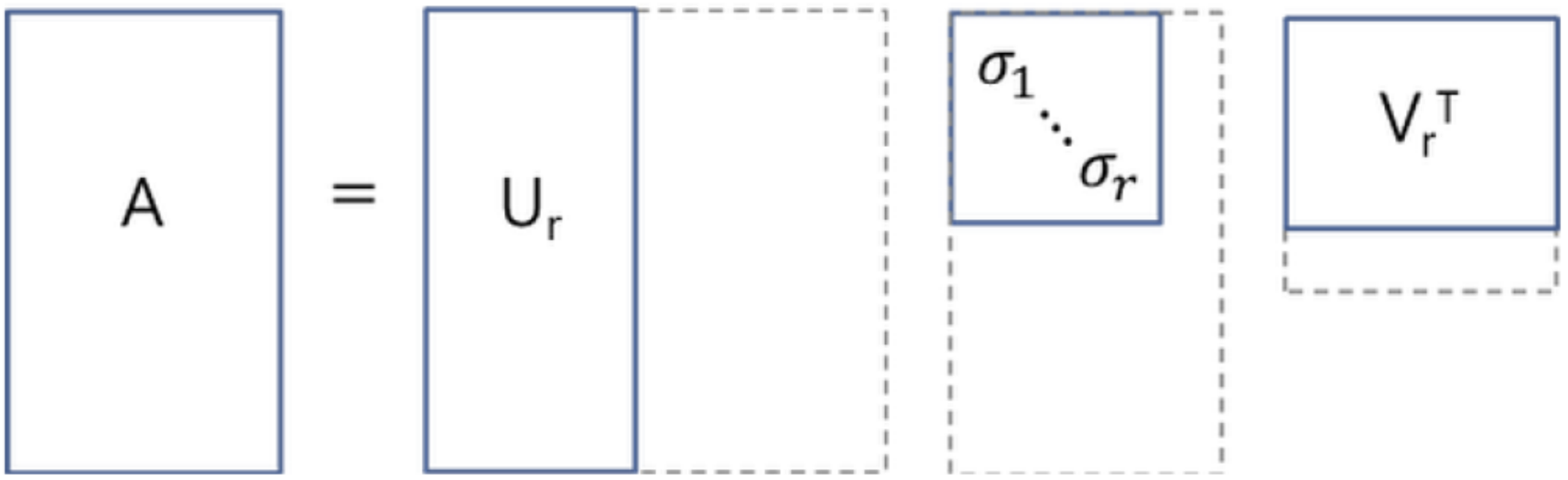


thin SVD



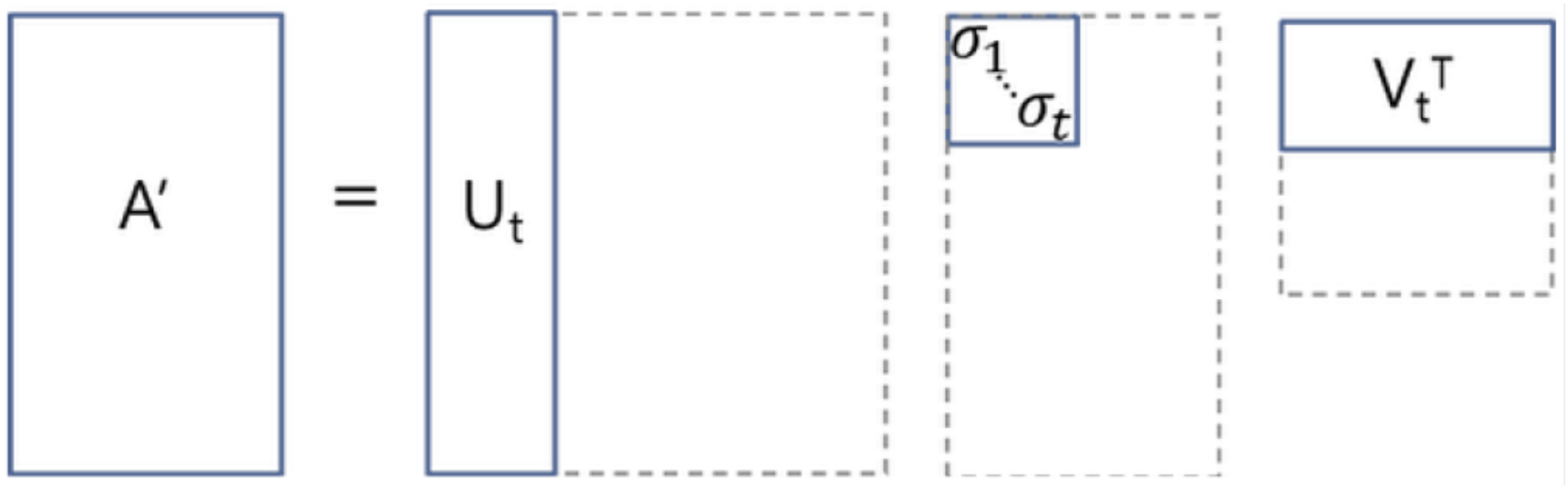
thin SVD는 Σ 행렬의 아랫부분(비대각 파트, 모두 0)과 U에서 여기에 해당하는 부분을 모두 제거한다. 이렇게 U와 Σ를 줄여도 UsΣsVT로 A를 원복할 수 있다.

compact SVD



compact SVD는 Σ 행렬에서 비대각파트뿐 아니라 대각원소(특이값)가 0인 부분도 모두 제거한 형태이다. 여기에 대응하는 U와 V의 요소 또한 제거한다. 다시 말해 특이값이 양수인 부분만 골라낸다는 뜻인데, 이렇게 U와 Σ, V를 줄여도 UrΣrVTr로 A를 원복할 수 있다.

truncated SVD

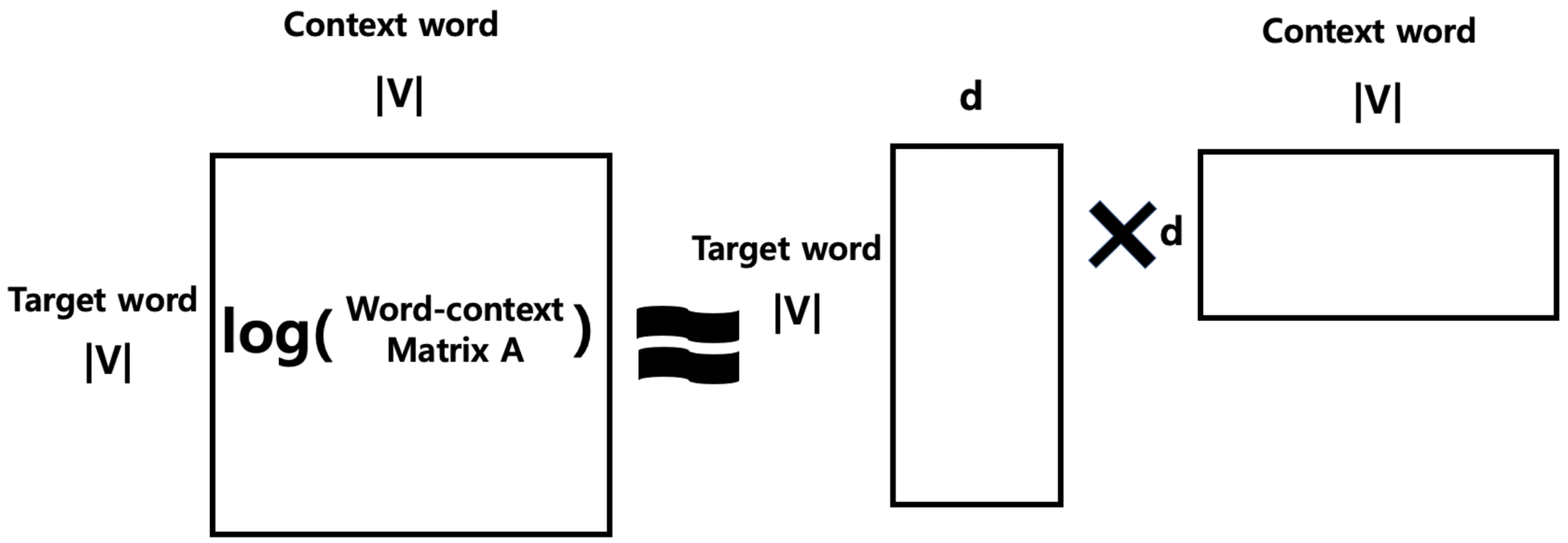


truncated SVD는 Σ 행렬의 대각원소(특이값) 가운데 상위 t개만 골라낸 형태이다. 이렇게 하면 행렬 A를 원복할 수 없게 되지만, 데이터 정보를 상당히 압축했음에도 행렬 A를 근사할 수 있게 된다.

LSA를 적용하면 단어와 문맥 간의 내재적인 의미를 효과적으로 보존할 수 있게 돼 결과적으로 문서 간 유사도 측정 등 모델의 성능 향상에 도움을 줄 수 있다고 한다. 또한 입력 데이터의 노이즈, 희소성을 줄일 수 있다.

GloVe

단어-문맥 행렬 분해를 통해 목적함수를 최소화하며 학습



목적함수

$$J = \sum_{i,j=1}^{|V|} f(A_{ij})(U_i \cdot V_j + b_i + b + j - logA_{ij})^2$$

bias항 2개와 f(A$_{ij}$)는 임베딩
품질을 높이기 위해 고안된 장치

**임베딩된 두 단어 벡터의 내적이 말뭉치 전체에서의
동시 증장 빈도의 로그 값이 되도록 정의한다.**

**단어 i, j 각각에 해당하는 벡터
U$_i$, V$_j$ 사이의 내적값과 두 단어 동시 등장 빈도
A$_{ij}$ 의 로그값 사이의 차이가 최소화될수록 학습 손실이 작아진다.**

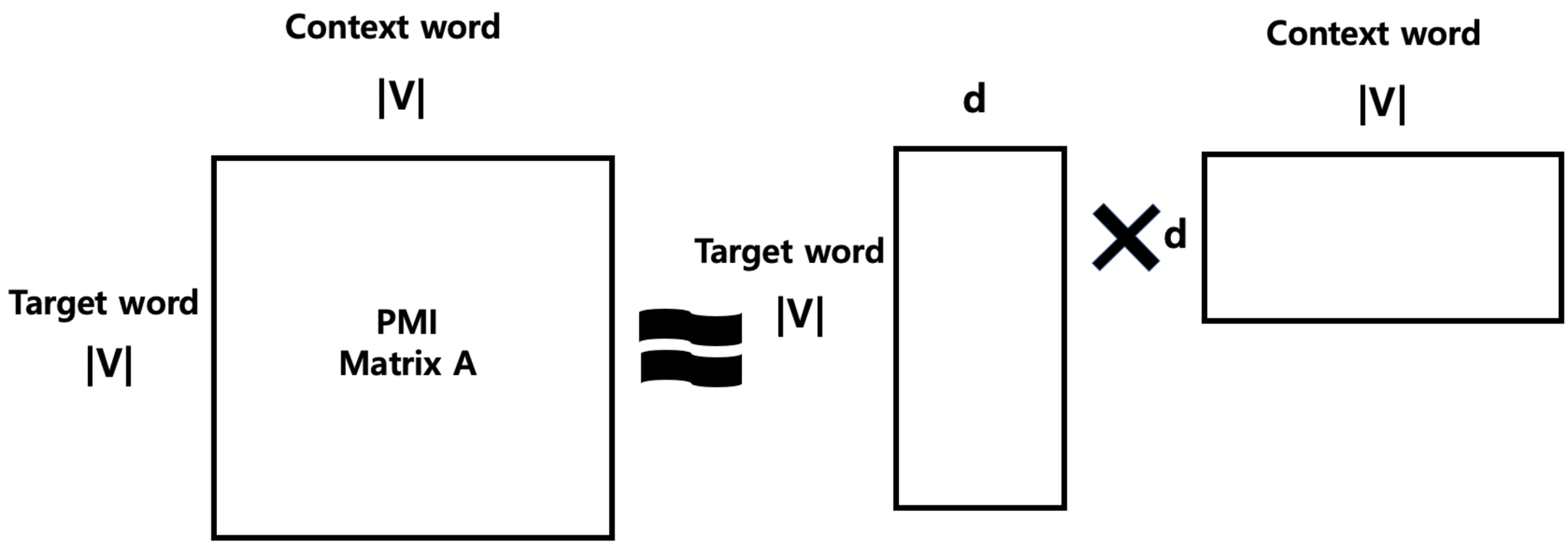
**Glove는 단어-문맥 행렬 A를 만든 후에 학습이
끝나면 U를 단어 임베딩으로 사용 가능**

LSA(잠재 의미 분석)은 Corpus 전체의 통계량을 모두 활용할 수 있지만, 그 결과물로 단어 간 유사도를 측정하기는 어렵다. 반대로 Word2Vec은 단어 벡터 사이의 유사도를 측정하는 데는 LSA보다 유리하지만 사용자가 지정한 window 내의 local context만 학습하기 때문에 Corpus 전체의 통계 정보는 반영되기 어렵다는 단점을 지닌다.

물론 GloVe 이후 발표된 Skip-gram 모델이 Corpus 전체의 Global한 통계량인 SPMI 행렬을 분해하는 것과 동치라는 점을 증명하기는 했다.

Swivel(Submatrix-Wise Vector Embedding Learner)

PMI 행렬 분해를 통해 목적함수를 최소화하며 학습



목적함수

$$x_{ij} > 0 \quad \frac{1}{2} f(x_{ij})(w_i^\top \tilde{w}_j - \mathbf{pmi}(i;j))^2$$

$$x_{ij} = 0 \quad \log\left[1 + \exp\left(w_i^\top \tilde{w}_j - \mathbf{pmi}^*(i;j)\right)\right]$$

PMI 행렬을 분해한다는 점에서 단어-문맥 행렬을 분해하는 GloVe와 다르며,
Swivel은 목적함수를 PMI의 단점을 보완할 수 있도록 설계했다.
두 단어가 한번도 동시에 등장하지 않았을 경우 PMI가 음의 무한대로 발산하는 현상을 보완하기 위해 손실함수를 [말뭉치가 동시 등장한 케이스가 한 건이라도 있는 경우/말뭉치에 동시 등장한 케이스가 한 건도 없는 경우]로 따로 정의했다.

그 결과, i, j가 각각 고빈도 단어인데 두 단어의 동시 등장빈도가 0이라면 두 단어는 정말로 등장하지 않는 의미상 무관계한 단어라고 가정하고, 단어 i, j가 저빈도 단어인데 두 단어의 동시 등장빈도가 0인 경우에는 두 단어는 의미상 관계가 일부 있을 수 있다고 가정한다.