Text Mining

presentation

TABLE OF - CONTENTS

TF-IDF란?

여러 개의 문서가 있을 때 각 문서내의 단어들에 가중치가 적용된 어떤 수치 값을 부여해 놓는 방법.

검색 키워드에 가장 부합하는 문서를 검색 결과 최상위에 배치하는데 기반이 되는 알고리즘.

TF-IDF란?

특정 단어의 중요도는 단어가 출현한 횟수에 **비례**하고 그 단어가 언급된 모든 문서의 총수에 **반비례**한다.

특정단어의 중요도 : TF(Term Frequency)

단어가 언급된 문서의 총수에 반비례 : IDF(Inverse Document Frequency)

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

$tf_{x,y}$ = frequency of x in y
df_x = number of documents containing x
N = total number of documents

TF-IDF란?

Keyword = “코코아”

TF = 30 TF = 0 TF = 10 TF = 35 TF = 0

Keyword = “코코아” DF = 3

Keyword = “날씨” DF = 5

Keyword = “사람” DF = 1

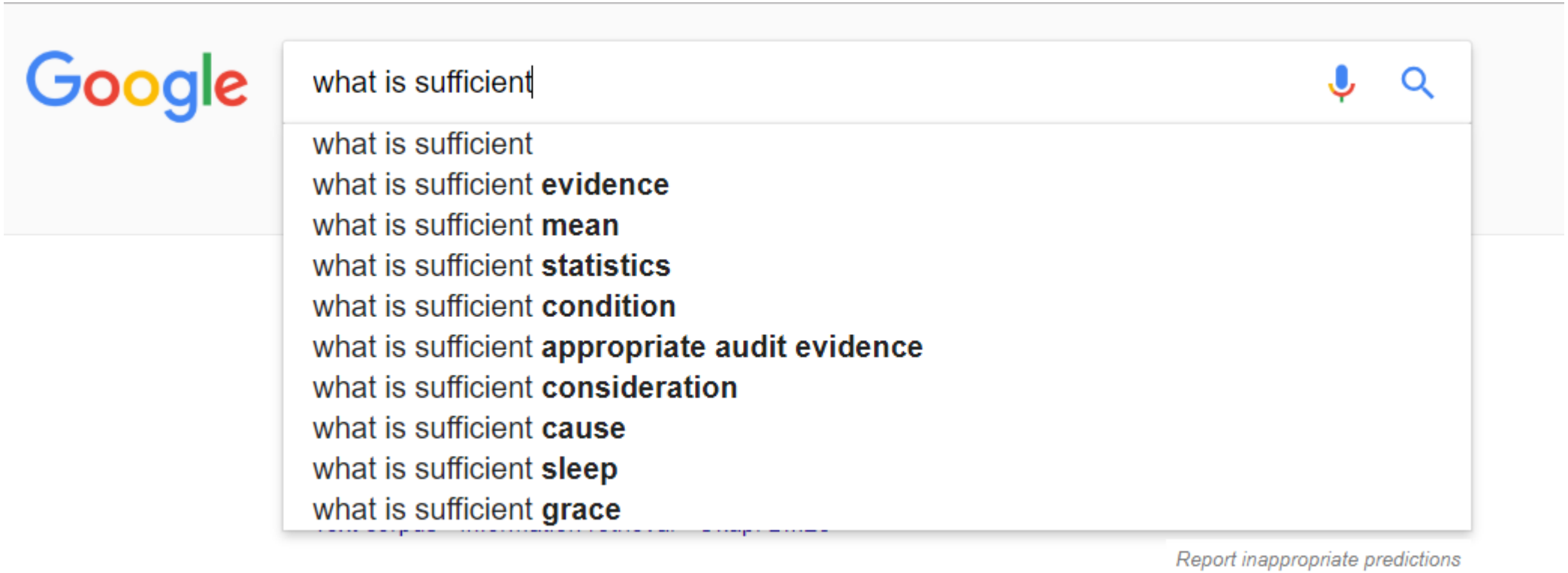
TF-IDF란?

뉴스 제목	이영학 1심 사형 선고..."정의의 이름으로 영원히 격리"	안전진단 막힌 재건축 5곳중 4곳 비강남...강남만 키운다	美 통상압박 시각차...與 "안보와 별개" vs 野 "동맹 균열"
TF-IDF	이영학, 2.995732273553991	강남, 4.127134385045092	미국, 2.797281334830153
TF-IDF	딸, 2.718121703031321	재건축, 2.860270868800483	안보, 1.8885939631132513
TF-IDF	혐의, 1.440023395335774	아파트, 2.355517186070986	정부, 1.4961070964471255
TF-IDF	피해자, 1.3770361751570304	안전, 2.292852436136162	트럼프, 1.259062642075501

TF-IDF가 높은 4개의 값을 추출하여 키워드 알아낼 수 있다.

부여된 점수 코사인 유사도(Cosine Similarity)를 이용하여 “문서의 유사도”를 구하는데도 쓰인다

N-gram이란?


Google
what is sufficient
what is sufficient
what is sufficient evidence
what is sufficient mean
what is sufficient statistics
what is sufficient condition
what is sufficient appropriate audit evidence
what is sufficient consideration
what is sufficient cause
what is sufficient sleep
what is sufficient grace
Report inappropriate predictions

N-gram이란?

N-gram은 주어진 배열에서 n개의 연속된 item들이다.

Input : The dog smelled like a skunk

Bigram : The dog, dog smelled, smelled like, like a, a skunk

Trigram : The dog smelled, dog smelled like, smelled like a, like a, a skunk

마르코프 체인(Markov Chain)

과거와 현재에 관한 정보가 주어질 때 미래에 관한 조건부분포는 현재의 정보에만 의존하며 과거의 정보와는 무관하다는 것이다.

$$P(X_{n+1} \in A | X_0 = x_0, \ldots, X_n = x_n) = P(X_{n+1} \in A) = P(X_{n+1} \in A | X_n = x_n)$$

다음 단어가 출현할 확률은 그전의 N-1개의 단어들로 결정된다.
(N-gram)

다음 단어가 출현할 확률은 그전의 N-1개의 단어들로 결정된다. (N-gram)

$$P(w_n|w_1 \dots w_{n-1}) = P(w_n|w_{n-k}\ w_{n-k+1} \dots w_{n-1})$$

(k+1)-gram or K^{th} order Markov approximation

Unigram: $P(w_1\ w_2 \dots w_n) = P(w_1)\ P(w_2) \dots P(w_n)$

Bigram: $P(w_1\ w_2 \dots w_n) = P(w_1)\ P(w_2|w_1) \dots P(w_n|w_{n-1})$

Trigram: $P(w_1\ w_2 \dots w_n) = P(w_1)\ P(w_2|w_1) \dots P(w_n|w_{n-2}\ w_{n-1})$

N -gram이란?

Estimate the likelihood of the sentence

I want to eat Chinese food.(Bigram)

P(I want to eat Chinese food) =
P(I | ⟨start⟩) * P*(to | want) * P(eat | to) *
P(Chinese | eat) * P(food | Chinese) *
P(⟨end⟩ | food)

N-gram이란?

Bigram	probability	Bigram	probability
Eat on	0.16	Eat Thai	0.03
Eat some	0.06	Eat breakfast	0.03
Eat lunch	0.06	Eat in	0.02
Eat dinner	0.05	Eat Chinese	0.02
Eat at	0.04	Eat Mexican	0.01
Eat Indian	0.04	Eat tomorrow	0.007
Eat today	0.03	Eat British	0.001

N-gram이란?

Bigram	probability	Bigram	probabilty
⟨start⟩ I	0.16	Want some	0.04
⟨start⟩ I'd	0.06	Want Thai	0.01
⟨start⟩ Tell	0.06	To eat	0.26
⟨start⟩ I'm	0.05	British food	0.60
I want	0.04	British restaurant	0.15
Want to	0.65	British cuisine	0.01
Want a	0.05	British lunch	0.01

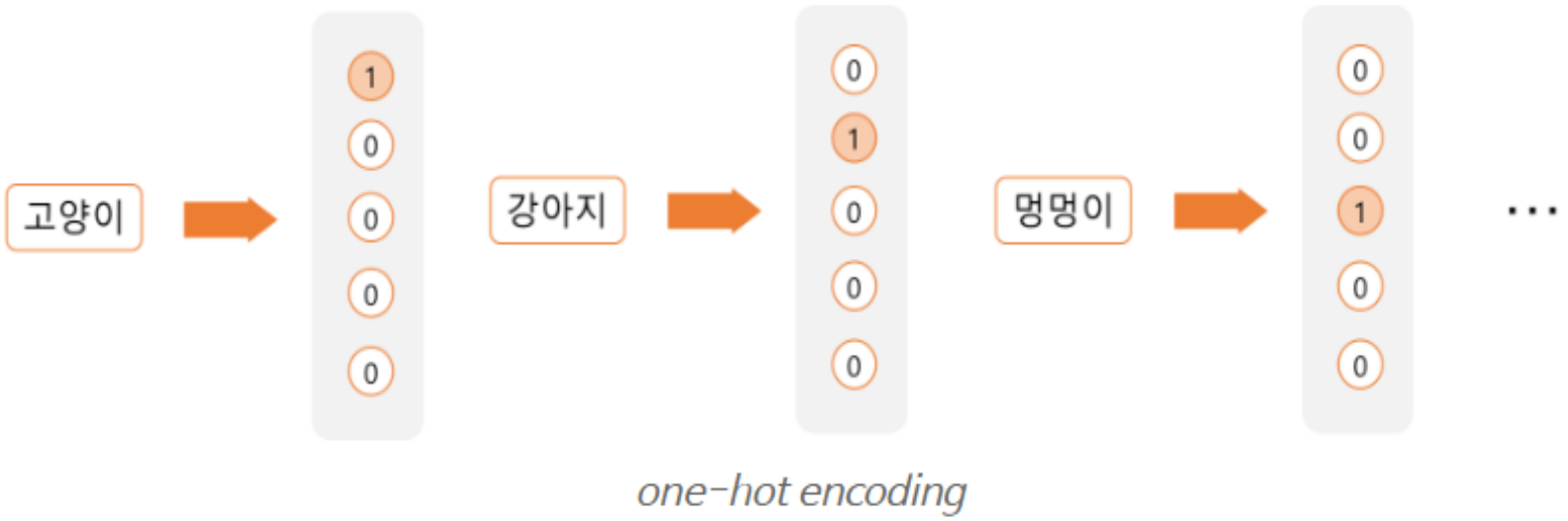
P(I want to eat Chinese food) =
P(I | ⟨start⟩) * P*(to | want) * P(eat | to) *
P(Chinese | eat) * P(food | Chinese) * P(⟨end⟩ | food)

=0.25*0.32*0.65*0.26*0.001*0.60 = 0.00008

N-gram의 N이 커질 경우 training corpus에서 해당되는 배열을 찾을 확률이 작다. 보통 Bigrams나 Trigrams를 사용한다. Large Corpus를 학습시키는 것이 중요

Word2Vec이란?

NLP(Natural Language Processing: 자연어처리



one-hot encoding

One Hot Encoding의 경우 단어와 단어간의 관계가 전혀 드러나지 않는다.

벡터에 단어의 의미를 담을 수는 없을까?

Word2Vec이란?

[초기모델]
-NNLM
-RNNLM

[최근]
-Word2vec
1)CROW
2)Skip-Gram

자연어 처리의 특성상 굉장히 많은 데이터를 넣어서 학습시켜줘야 하기 때문에
학습을 빠르게 할 수 있는 Word2vec이 주로 사용 되고 있다.

Word2Vec이란?

다음 단어를 예측하기 위해 Word Embedding을 구현해야 하는데 이때 1)CBOW or 2) Skip-Gram알고리즘을 사용

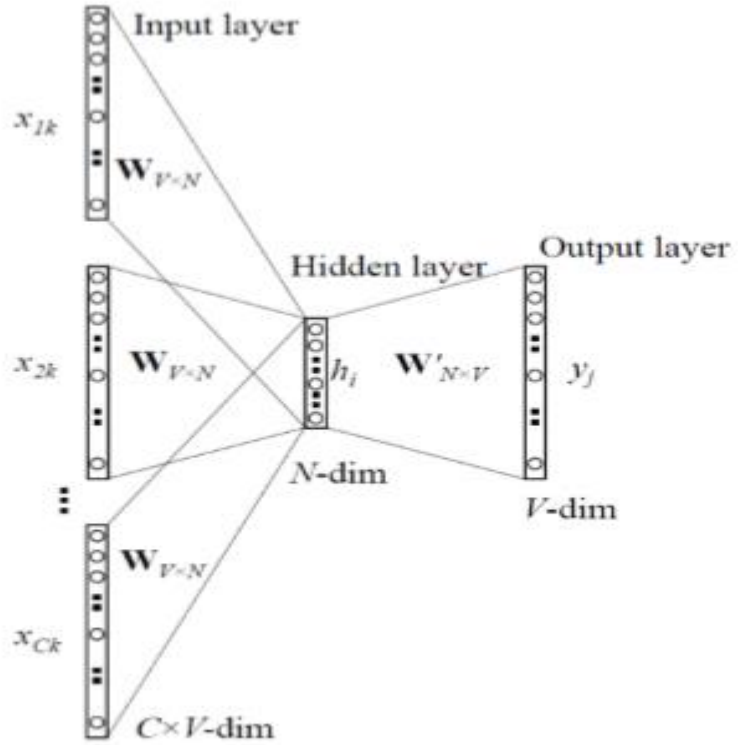
[특징1] 추론(예측) 가능

변환된 벡터가 단순한 수학적 존재 이상의 복잡한 개념 표현을 넘어 추론까지도 쉽게 구현 가능

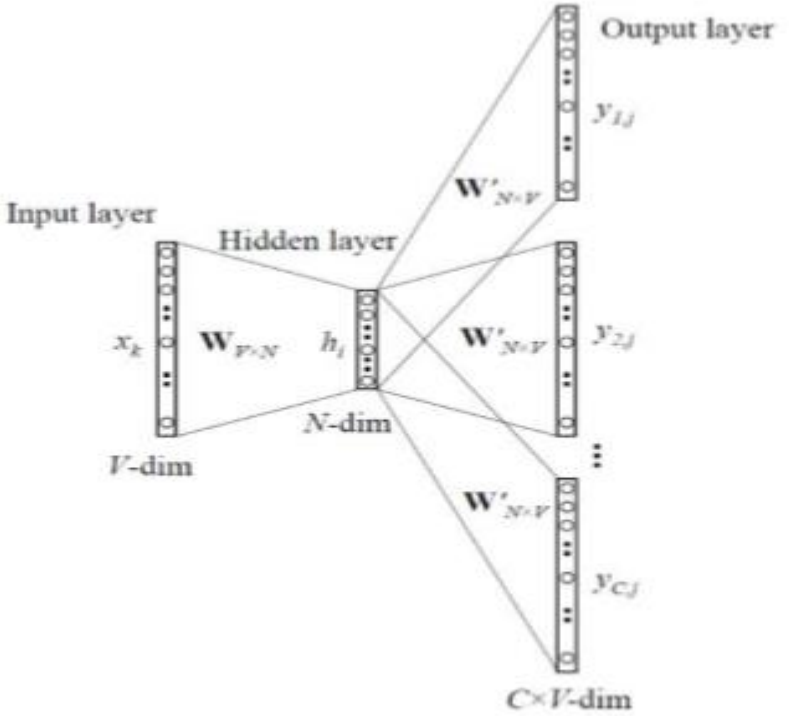
[특징2] 계산량의 획기적인 감소

기존 Neural Net기반 학습방법에 비해 크게 달라진 것은 아니지만 계산량을 엄청나게 줄여서 기존의 방법에 비해 몇 배 이상 빠른 학습이 가능

Word2Vec이란?



CBOW embedding



Skip Gram embedding

Word2Vec이란?

-Single -word context: '하나'의 단어로부터 다음에 오는 단어를 예측
-Multi-word context: '여러 개'의 단어를 나열, 그에 기반하여 단어를 추정

-context: 특정 단어 주변에 오는 단어들의 집합
'계산이 이루어지는 단어들'을 의미
Ex) the cat sits on the 에서 sits 양쪽 2개 단어 포함 총 5개의 단어가 컨텍스트가 되는 것
(목표 단어 양쪽 2개 단어만 허용한 경우, 허용 단어 개수는 임의로 지정)

1) CBOW(Contunuous Bag-of-words) 모델

컨텍스트로부터 찾고자 하는 목표 단어를 예측하는 모델
ex) the cat sits on the(컨텍스트) -〉 "mat"(목표단어)

2) Skip-Gram 모델

현재 하나의 단어를 통해 주변 단어들(컨텍스트)를 예측하는 모델
ex) the cat sits on the(컨텍스트) 〈- "mat"(현재단어)

CBOW모델

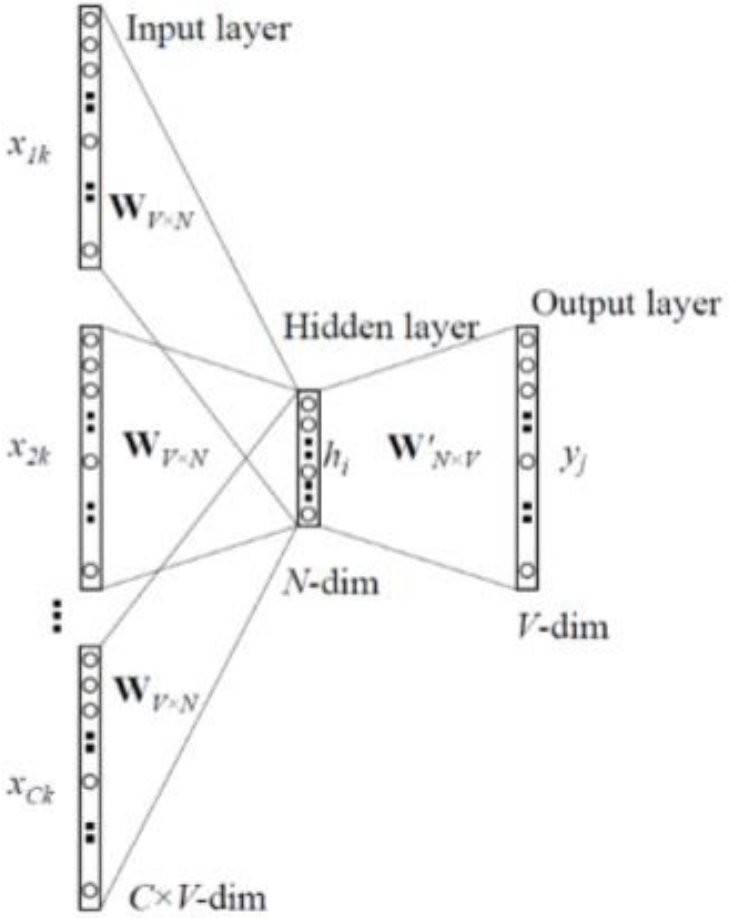
-CBOW(continuous Bag of words)모델

컨텍스트로부터 찾고자 하는 목표 단어를 예측하는 모델
Ex) the cat sits on the (컨텍스트) -〉 “mat”
(목표단어)

“아이스크림이 너무 __먹을 수 없었다”라는 문장에서 누구나 생략된 __부분의 단어를 추측할 수 있고 대부분은 옳게 예측한다. CBOW모델도 마찬가지의 방법을 사용한다.

– 주어진 단어에 대해 앞 뒤로 N/2개 씩 총 N개의 단어를 입력으로 사용하여 주어진 단어를 맞추기 위한 네트워크를 만든다

– 크기가 작은 데이터셋에 적합

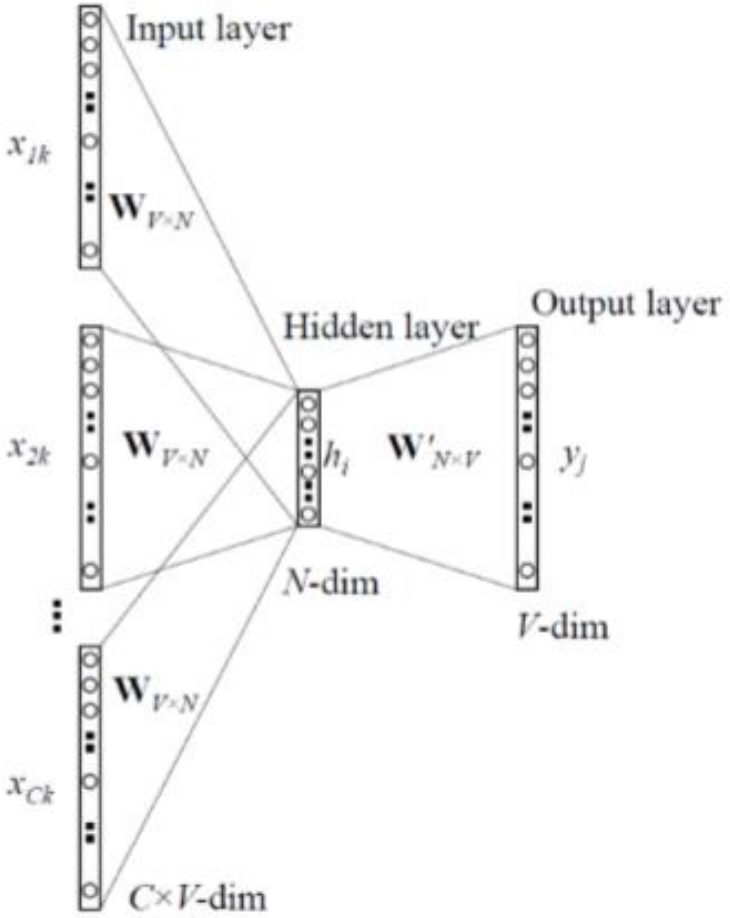


CBOW모델

“the quick brown fox jumped over the lazy dog”

The quick brown(컨텍스트 주어짐) –〉 fox(예측)

각 문맥 단어를 Hidden layer로 투사하는 가중치 행렬 (Wvn)은 모든 단어(X1k … Xck)에 공통으로 사용



YELLOW

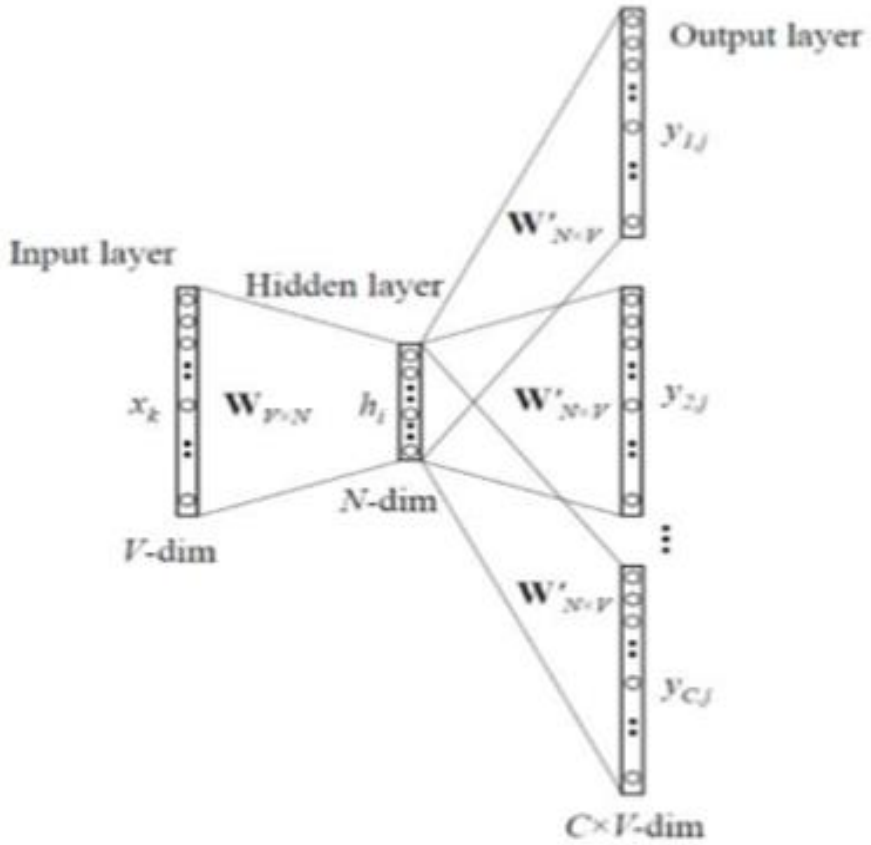
Skip-Gram모델

현재 하나의 단어를 통해 주변 단어들(컨텍스트)를 예측하는 모델

Ex) the cat sits on the (컨텍스트) <- “mat”
(현재단어)

일반적으로 입력 단어 주변 k개의 단어를 컨택스트로 보고 예측 모형을 만듬

예측하는 단어들은 현재 단어 주변에서 샘플링하는데, “가까이 있는 단어일수록 현재 단어와 관련이 더 많다”는 원리를 적용하기 위해 멀리 떨어진 단어를 낮은 확률로 선택하는 방법을 사용한다
나머지는 CBOW모델과 방향만 반대이고 거의 비슷하다.



크기가 큰 데이터셋에 적합
최근일수록 더욱 많은 데이터를 갖고 있기 때문에 주로 Skip Gram모델을 사용

Skip-Gram모델

the quick brown fox jumped over the lazy dog

1. 컨텍스트 정의
   일반적으로 구문론적(형식적) 컨텍스트를 정의
   일단 컨텍스트를 현재 단어(목표단어)의 양쪽에 있는 단어들의 윈도우로 정의해보자

2. Window size 설정 및 데이터셋 도출
   윈도우를 1로 하면 다음과 같은 쌍으로 구성된 데이터셋이 도출됨

([The, brown],quick), ([quick,fox], brown),([brown, jumped],fox),...

3. 목적: 'quick'이라는 현재 단어로부터 컨텍스트 'the'와 'brown'을 예측!
   아래와 같은 데이터셋(단어 관계)을 도출할 수 있어야됨

(quick, the), (quick, brown), (brown, quick), (brown, fox),…

Thank you。