

Enriching Word Vectors with Subword Information

Piotr Bojanowski* and **Edouard Grave*** and **Armand Joulin** and **Tomas Mikolov**
Facebook AI Research
`{bojanowski, egrave, ajoulin, tmikolov}@fb.com`

이준영

UST-KRISS 측정과학전공

2021. 03. 26

자연어 처리는 한 단어의 의미는 주변 단어에 의해 형성된다는 **분포가설**에 기반해 발전해왔다.

라벨링 되지 않은 대용량의 말뭉치로부터 단어의 분산표현을 얻기 위해 사용된 신경망은 **word2vec**의 **CBOW**와 **skip-gram**모델이 대표적이다.

word2vec의 분산표현은 데이터를 처리하는 가장 작은 단위가 어휘이기 때문에 다음과 같은 한계를 갖고있다.

1. 같은 의미이지만 문장구조에 따라 다른 형태를 갖는 특성이 고려되지 못한다.
(ex. eat – eats – eaten – eater - eating)
2. 학습 데이터에 없는 단어의 임베딩이 불가능하다. (OOV, Out Of Vocabulary)

본 논문은 데이터를 처리하는 가장 작은 단위를 '어휘'가 아닌 '**어휘를 구성하는 글자 n-gram**'으로 한 단계 낮추는 것을 제안한다. 즉, 어휘의 하위 단계 (subword)인 n-gram 벡터의 합으로 어휘를 임베딩 하는 것이다.

Skip-gram

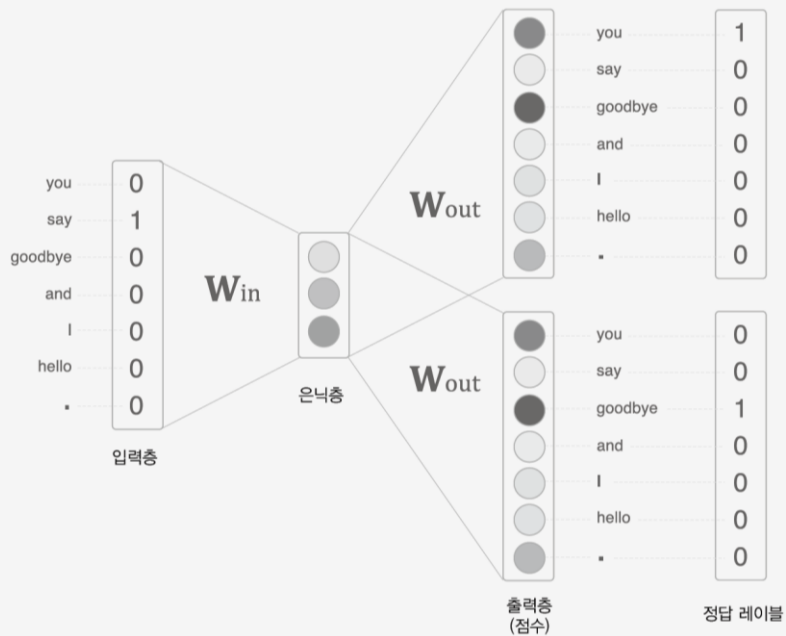
+

Negative Sampling

+

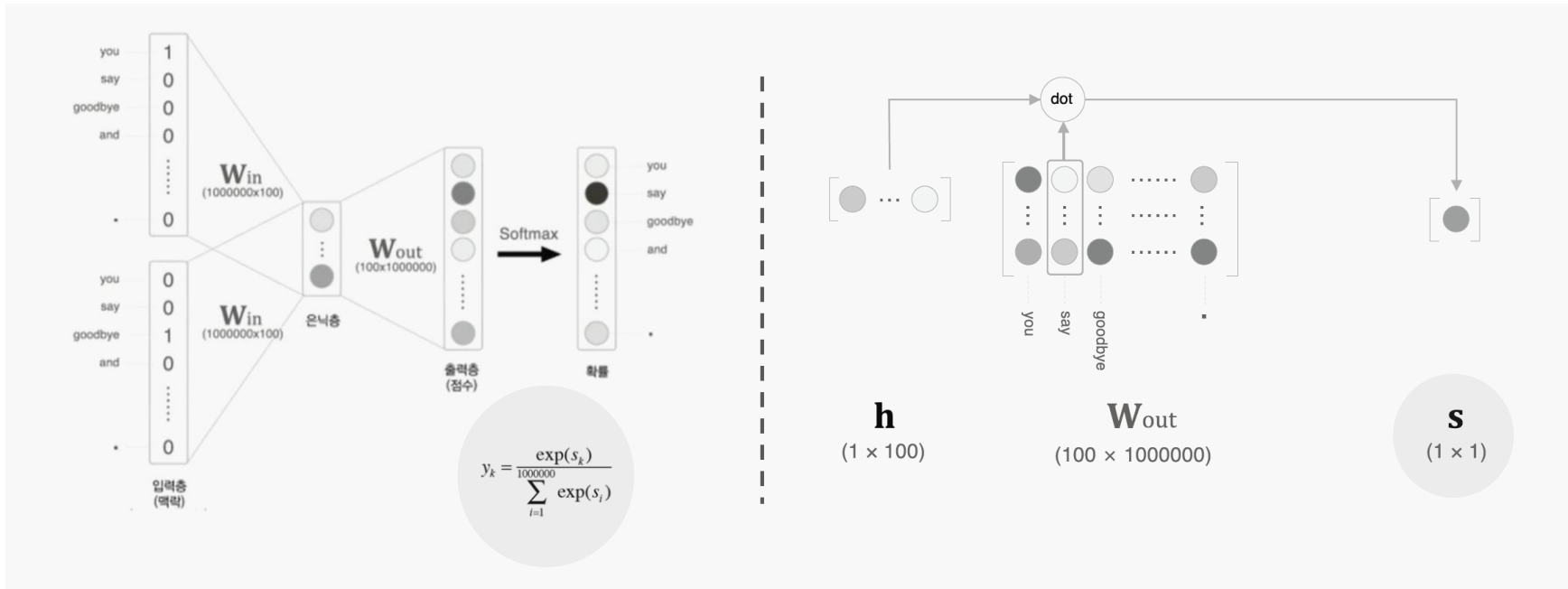
Character n-gram

3.1 General model: skipgram model



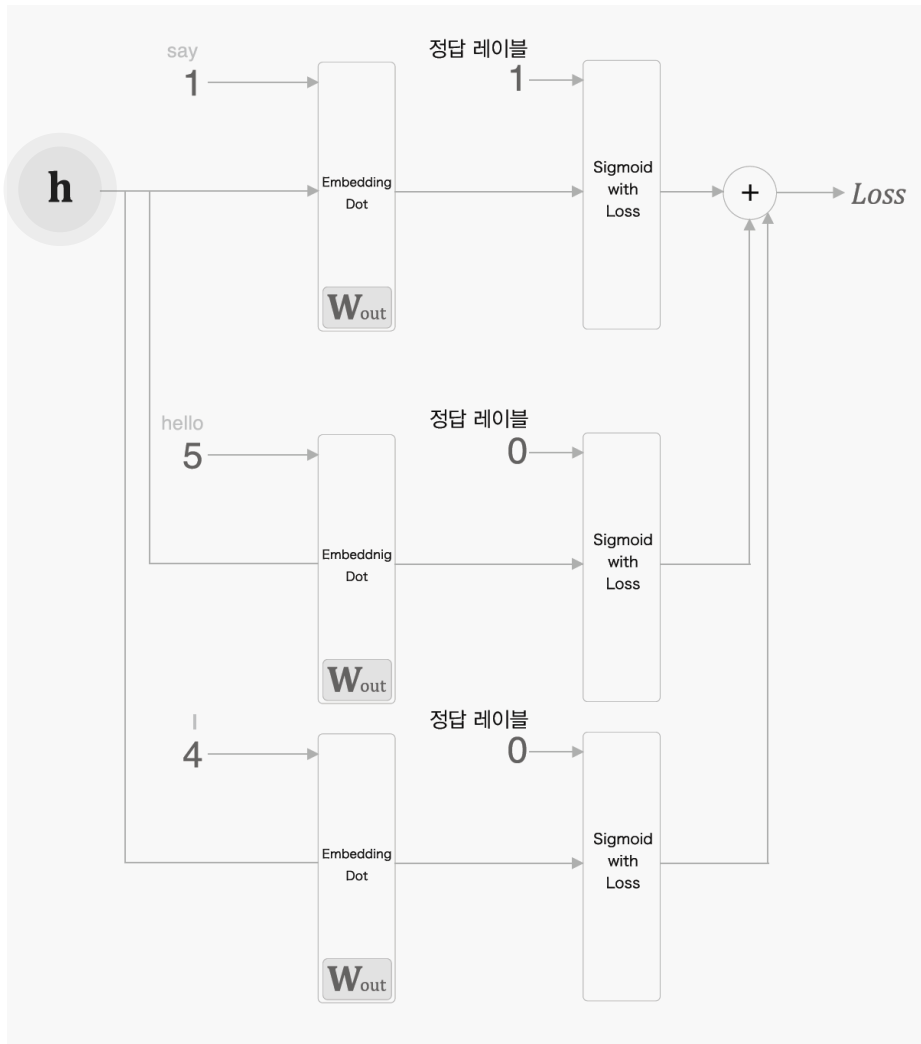
Window Size	Text	Skip-grams
2	[The wide road shimmered] in the hot sun.	wide, the wide, road wide, shimmered
	The [wide road shimmered in the] hot sun.	shimmered, wide shimmered, road shimmered, in shimmered, the
	The wide road shimmered in [the hot sun].	sun, the sun, hot
3	[The wide road shimmered in] the hot sun.	wide, the wide, road wide, shimmered wide, in
	[The wide road shimmered in the hot] sun.	shimmered, the shimmered, wide shimmered, road shimmered, in shimmered, the shimmered, hot
	The wide road shimmered [in the hot sun].	sun, in sun, the sun, hot

3.1 General model: skipgram model + Negative Sampling



- 어휘수 100만개의 경우, Softmax 계산을 위해서 **exponential계산**을 1,000,000번 수행해줘야 함.
- 네거티브 샘플링의 핵심은 **단어 하나에 대한 '이진 분류'** binary classification'에 있음.
(맥락이 'you', 'good bye'일 때, 타깃 단어는 'say'인가? 에 대한 yes / no)
-
- **Score** 값은 은닉층과 출력 가중치의 embedding의 내적이 됨.

3.1 General model: skipgram model + Negative Sampling



- 네거티브 샘플링 기법은 맥락에 대한 분산벡터 계산 후, 긍정적인 예를 타깃으로 한 경우와 부정적인 예 몇가지를 샘플링한 경우에 대한 이진 분류의 손실을 계산하여 모두 더한 값을 최종 손실 값으로 함.

3.1 General model: skipgram model + Negative Sampling

$$w \in \{1, \dots, W\}$$

$$w_1, \dots, w_T$$

$$\sum_{t=1}^T \sum_{c \in \mathcal{C}_t} \log p(w_c | w_t) = \frac{e^{s(w_t, w_c)}}{\sum_{j=1}^W e^{s(w_t, j)}}$$

문맥단어 타겟단어

문맥단어 세트 소프트맥스 함수

3.1 General model: **skipgram model + Negative Sampling**

이진분류의
손실함수


$$\log \left(1 + e^{-s(w_t, w_c)} \right) + \sum_{n \in \mathcal{N}_{t,c}} \log \left(1 + e^{s(w_t, n)} \right)$$

Skipgram의
손실함수

$$\sum_{t=1}^T \left[\sum_{c \in \mathcal{C}_t} \ell(s(w_t, w_c)) + \sum_{n \in \mathcal{N}_{t,c}} \ell(-s(w_t, n)) \right]$$
$$s(w_t, w_c) = \mathbf{u}_{w_t}^\top \mathbf{v}_{w_c}, \quad \ell : x \mapsto \log(1 + e^{-x})$$

3.2 Subword model

character n-grams

where  <wh, whe, her, ere, re>, <where>

Word	Length(n)	Character n-grams
eating	3	<ea, eat, ati, tin, ing, ng>
eating	4	<eat, eati, atin, ting, ing>
eating	5	<eati, eatin, ating, ting>
eating	6	<eatin, eating, ating>

$$s(w, c) = \sum_{g \in \mathcal{G}_w} \mathbf{z}_g^\top \mathbf{v}_c$$

$$\mathcal{G}_w \subset \{1, \dots, G\}$$

3.2 Subword model



- C++로 구현, CBOW와 skip-gram을 기준으로 삼음.
- linear decay of the step size
- word vector의 차원: 300
- context의 window size C: 1~5 중 랜덤하게 사용
- subsampling threshold: 10^{-4}
- 어휘는 데이터 내에 5회 이상 등장하는 단어만 사용
- 위키피디아를 데이터 셋으로 활용

(Arabic, Czech, German, English, Spanish, French, Italian, Romanian and Russian)

5.1 Human similarity judgement (Spearman's rank correlation)

: Human similarity judgement & cosine similarity between vector representation

		sg	cbow	sisg-	sisg
AR	WS353	51	52	54	55
	GUR350	61	62	64	70
DE	GUR65	78	78	81	81
	ZG222	35	38	41	44
EN	RW	43	43	46	47
	WS353	72	73	71	71
ES	WS353	57	58	58	59
FR	RG65	70	69	75	75
RO	WS353	48	52	51	54
RU	HJ	59	60	60	66

sisg : subword information skip-gram
OOV도 n-gram의 sum으로 표현

sisg- : OOV를 null vector로 사용

- OOV를 n-gram으로 표현했을 때 효과가 잘나타남
- 문법적 기능에 따라 형태가 변하는 단어나, 합성어가 많은 독일어, 러시아어에서 상대적으로 효과가 큼
- English WS353은 rare word가 적은 데이터셋이었기 때문에 큰 효과는 없었음.

5.2 Word analogy task

		sg	cbow	sisg
CS	Semantic	25.7	27.6	27.5
	Syntactic	52.8	55.0	77.8
DE	Semantic	66.5	66.8	62.3
	Syntactic	44.5	45.0	56.4
EN	Semantic	78.5	78.2	77.8
	Syntactic	70.1	69.9	74.9
IT	Semantic	52.3	54.7	52.3
	Syntactic	51.5	51.8	62.7

metric : accuracy

- 구문분석에는 좋음

Singular/plural Base/Comparative
 cat → cats good → better
 dog → ? rough → ?

- 의미분석에는 효과가 없음
 → 적절한 n-gram 설정으로 성능향상 가능

man → king
 woman → ?

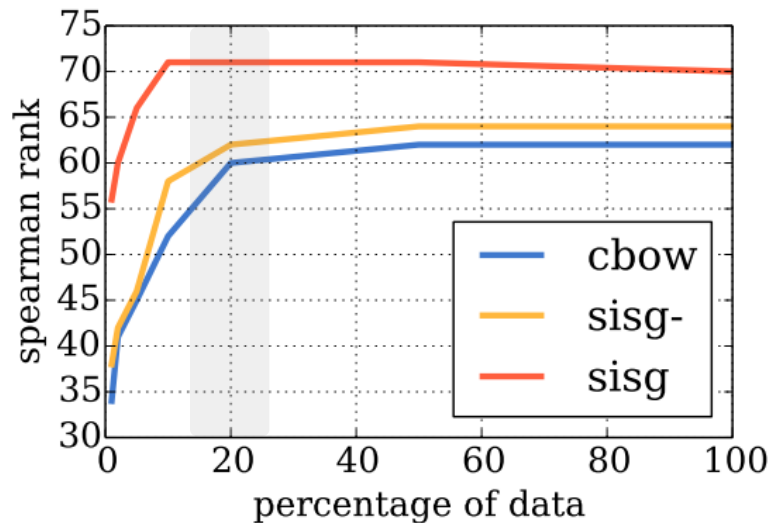
5.3 Comparison with morphological representations

(Spearman's rank correlation coefficient)

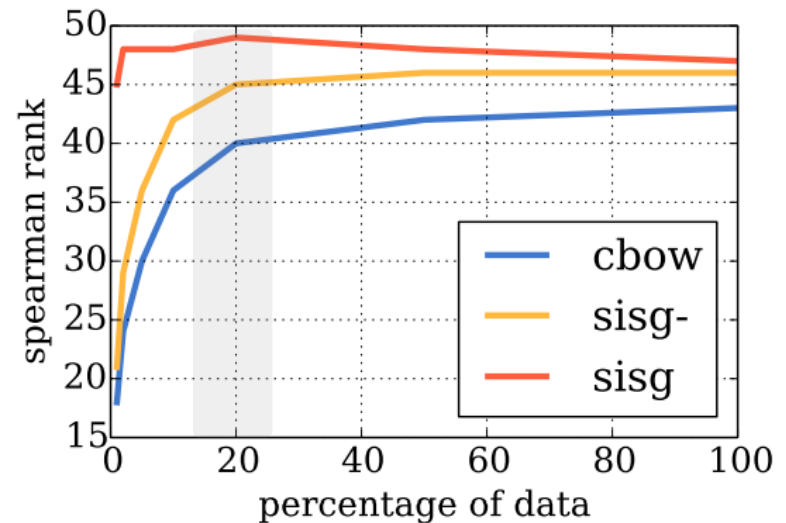
	DE		EN		ES	FR
	GUR350	ZG222	WS353	RW	WS353	RG65
Luong et al. (2013)	-	-	64	34	-	-
Qiu et al. (2014)	-	-	65	33	-	-
Soricut and Och (2015)	64	22	71	42	47	67
sisg	73	43	73	48	54	69
Botha and Blunsom (2014)	56	25	39	30	28	45
sisg	66	34	54	41	49	52

- 다른 형태론적 접근모델과 비교하여 더 높은 성능을 나타냄
- 다른 모델은 합성어와 같은 복합적인 단어는 제대로 모델링하지 못함.

5.4 Effect of the size of the training data



(a) DE-GUR350



(b) EN-RW

- 모든 데이터 크기에 대해 sisg가 우수한 성능을 보임
- 데이터의 크기의 증가가 성능 증가에 무한히 비례하지는 않지만,
- Sisg는 적은 데이터셋만으로도 충분한 성능을 보임.

		b				
		2	3	4	5	6
a	2	57	64	67	69	69
	3		65	68	70	70
	4			70	70	71
	5				69	71
	6					70
	(a) DE-GUR350					
	2	3	4	5	6	
	2	59	55	56	59	60
	3		60	58	60	62
	4			62	62	63
	5				64	64
	6					65
(b) DE Semantic						
	2	3	4	5	6	
	2	45	50	53	54	55
	3		51	55	55	56
	4			54	56	56
	5				56	56
	6					54
(c) DE Syntactic						
	2	3	4	5	6	
	2	41	42	46	47	48
	3		44	46	48	48
	4			47	48	48
	5				48	48
	6					48
(d) EN-RW						
	2	3	4	5	6	
	2	78	76	75	76	76
	3		78	77	78	77
	4			79	79	79
	5				80	79
	6					80
(e) EN Semantic						
	2	3	4	5	6	
	2	70	71	73	74	73
	3		72	74	75	74
	4			74	75	75
	5				74	74
	6					72
(f) EN Syntactic						

$$a \leq n \leq b$$

- 16

5.6 Language modeling

	Cs	DE	Es	FR	RU
Vocab. size	46k	37k	27k	25k	63k
CLBL	465	296	200	225	304
CANLM	371	239	165	184	261
LSTM	366	222	157	173	262
sg	339	216	150	162	237
sisg	312	206	145	159	206

LSTM : 기본 LSTM (baseline)

sg : LSTM + skip gram

sisg : LSTM + skip gram + n-gram

- 제안된 모델의 성능이 제일 좋음.
- 형태학적으로 풍부한 언어에 대해 특히 더 두드러짐.

- 단어의 분산표현을 구하기 위해, **character n-gram**과 **skip gram**을 함께 사용하는 것을 제안.
- 제안된 방법을 통해 **데이터 전처리나 지도과정 없이** 텍스트 데이터를 빠르고 간단하게 분산표현을 구할 수 있음

Q&A