논문 요약

날짜	@2021년 3월 12일 → 2021년 3월 15일
배정	상민 이㉿ 유경 한 KM S Yujin Kim 예진 문
상태	완료

Enriching Word Vectors with Subword Information

> Piotr Bojanowski∗ and Edouard Grave∗and Armand Joulin and Tomas Mikolov
> Facebook AI Research

1. Abstract & Introduction

1. **Existing problems**

   → ignore the morphology of words

   → assigning a distinct vector to each word

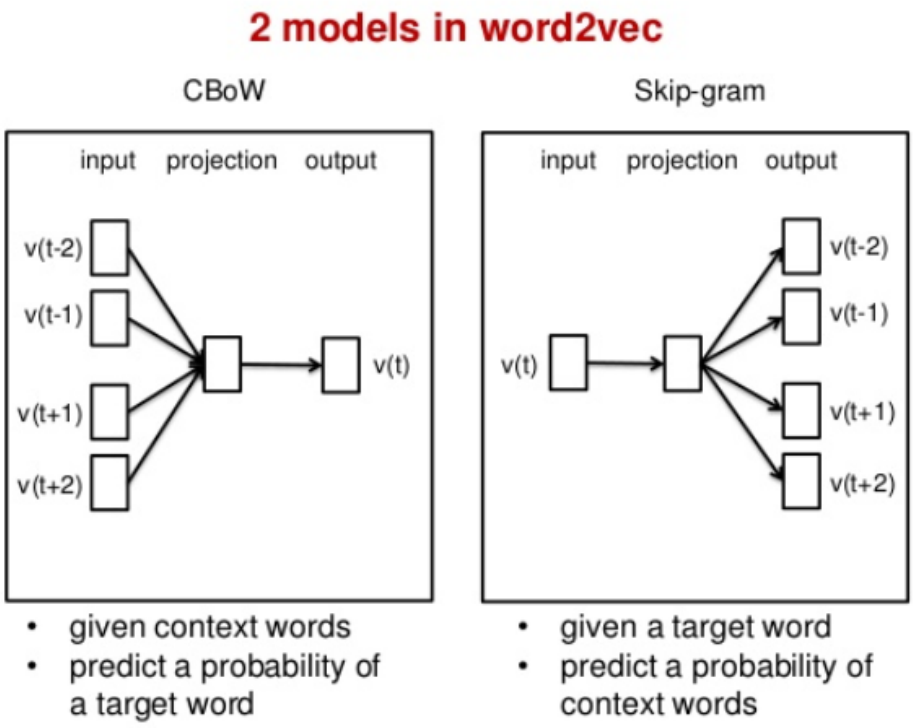
   ex. eat과 eating에 각각 distinct vector를 부여 → 단어간의 유사성 알 수 없음

   → limitation for languages with many rare words

   W2V과 같은 기존의 방법들은 distribution hypothesis 기반으로 학습하기 때문에 출현 횟수가 많은 단어에 대해서는 잘 embedding 되지만, 출현횟수가 적은 단어에 대해서는 제대로 embedding 되지 않음

2. **Proposal**

   → based on the skip gram model(each word is represented as a bag of character n-grams)



ex. CBoW : 나는 향긋한 _____를 좋아한다

Skip-gram : _____ 외나무다리에서 _____

→ vector representation is associated to each character n-gram

→ words being represented as the sum of these representations

3. **Contribution**

   → fast - train models on large corpora quickly

   → compute word representations for words that did not appear in the training data

   ex. eat 학습 → eating 추론 가능 & orange 학습 → oranze 추론가능 : oov 문제 해결

4. **State-of-the-art performance !**

2. Related work

1. **Morphological word representations** - incorporate morphological information into word representations.

→ Alexandrescu and Kirchhoff (2006) :

introduced factored neural language models, where words are represented as sets of features. These features might include morphological information. - succesfully applied to morphologically rich languages, such as Turkish (Sak et al., 2010).

→ [Lazaridou et al., 2013; Luonget al., 2013; Botha and Blunsom, 2014; Qiu etal., 2014] :

proposed different composition functions to derive representations of words from morphemes, rely on a morphological decomposition of words

→ Chen et al. (2015) :

introduced a method to jointly learn embeddings for Chinese words and characters.

→ Cui et al. (2015) :

proposed to constrain morphologically similar words to have similar representations. \

→ Soricut and Och (2015) :
described a method to learn vector representations of morphological transformations, allowing to obtain representations for unseen words by applying these rules.

→ Cotterell and Schütze (2015) :

introduce Word representations trained on morphologically annotated data

→ Schütze (1993) : *closest to fast text approach!*

learned representations of character four-grams through singular value decomposition,
and derived representations for words by summing the four-grams representations.

→Wieting et al. (2016) :

proposed to represent words using character n-gram count vectors. However, the objective function used to learn these representations is based on paraphrase pairs, while fast text model can be trained on any text corpus.

2. **Character level features for NLP -** character level models for natural language processing

   These models discard the segmentation into words and aim at learning language representations directly from characters.

   (1) Recurrent neural networks applied to ..

   → **[Mikolov et al., 2012; Sutskever et al., 2011; Graves, 2013; Bojanowski et al., 2015]** :

   language modeling

   → **Chrupała, 2014 :** text normalization

   → **Ling et al., 2015 :** part-of-speech tagging

   → **Ballesteros etal., 2015 :** parsing

   (2) Another family of models are convolutional neural networks trained on characters, which
   were applied to ...

   → **dos Santos and Zadrozny, 2014 :** part-of-speech tagging

   → **dos Santos and Gatti, 2014** : sentiment analysis

   → **Zhang et al., 2015 :** text classification

   → **[Kim et al.,2016). Sperr et al. (2013)]** : language modeling

   introduced a language model based on restricted Boltzmann machines, in which words are encoded as a set of character n-grams.

   (3) Recent works in machine translation

   → **Sennrich et al., 2016; Luong and Manning, 2016 :**

   using subword units to obtain representations of rare words

3. Model

Model to learn word representations while taking into account morphology by...

- considering subword units
- representing words by a sum of its character n-grams

3.1 General Models

— Skip - gram : to learn a vectorial representation for each word w

- given a word vocabulary of size W
- word is identified by its index w ∈ {1, ..., W}
- train to predict well words that appear in its context (inspired by the distributional hypothesis)

총 W개의 단어들에 각각 벡터를 할당하고, 주변에 어떤 단어가 올지 예측하는것이 목표

- given a large training corpus represented as a sequence of words w1, ..., wT

  → skip - gram model is to maximize the following log-likelihood :

$$\sum_{t=1}^{T} \sum_{c \in \mathcal{C}_t} \log p(w_c \mid w_t)$$

wt라는 단어가 주어졌을 때, 어떤 문맥 단어 wc가 올 확률이 가장 높은가를 예측한다.
해당 목적함수에서 결과적으로 구해야 하는것은 p(wc | wt) 값.

— To calculate p(wc | wt)...

단어 : 고유벡터

Wt : Uwt

Wc : Uwc

이 두 벡터 (Uwt, Uwc) 를 가지고 확률값 p(wc | wt) 을 만들어야 함

→ One possible choice to define the probability of a context word is the softmax:

$$p(w_c \mid w_t) = \frac{e^{s(w_t,\, w_c)}}{\sum_{j=1}^{W} e^{s(w_t,\, j)}}.$$

$$= \frac{Wt가수어셨늘때,실제성납값Wc가수어실왁률}{Wt가주어졌을때,다른모든단어들이등장할확률의합}$$

→ However, such a model is not adapted to our case as
it implies that, given a word wt, we only predict one context word wc.

주변 문맥단어 중 하나만을 제대로 예측할 수 있고, 나머지는 제대로 예측하지 못함

ex) O(Wc) X X Wt X X
X O(Wc) X Wt X X
X X O(Wc) Wt X X ...

→ Instead of softmax function, the model is trained by negative log-likelihood :

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

*negative sampling :정답이 아닌것들 중 랜덤으로 추출

*s(wt, wc) : score를 구하는 함수, wt와 wc를 내적한다. wc가 실제 정답일 경우 두 벡터를 유사하게(similarity가 높게) 계산함

*s(wt, n) : 실제 정답이 아닌 단어 n 에 관해서는 similarity를 낮게 계산함

3.2 Subword model :

→ By using a distinct vector representation for each word, the skip-gram model ignores the internal structure of words

ex. 기존의 distinct vector representation은 teach, teacher, teachers등의 단어들의 내부 구조가 동일 하다는 것을 알 수 없음

→ We propose a different scoring function s, in order to take into account the internal structure of words.

- add special boundary symbols < and > at the beginning and end of words
- each word w is represented as a bag of character n-gram

- distinguish prefixes and suffixes from other character sequences.
- also include the word w itself in the set of its n-grams, to learn a representation for each word
- represent a word by the sum of the vector representations of its n-grams.

→We thus obtain the scoring function:

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

g : n-gram dictionary size

gw : set of n-grams appealing in word w (단어 w에서 나올 수 있는 모든 n-gram의 세트)

zg : vector representation to each n_gram g

ex. character n - gram for word *where* and n = 3 :

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

→subword model은 n =1 , n= 2, n = 3, n = 4 각각의 n-gram으로 부터 도출된 벡터를 모두 더한것을 단어 *where*의 벡터로 표현

→This simple model allows ...

- sharing the representations across words, thus allowing to learn reliable representation for rare words.

4. Experimental setup

4.1 Baseline

- Word2Vec 패키지에 있는 skipgram 모델, cbow 모델(C언어 베이스)을 우리 모델과 비교 → 단어표현 계산

4.2 Optimization

- Negative log likelihood(binary cross entropy)를 SGD로 최적화
- skipgram 모델: 선형으로 감소하는 step-size 사용
- Training set

  1) T = word 개수

  2) P = pass 수 / data

  3) step size = $\gamma_0(1 - \frac{t}{TP})$: @t, 상수 γ_0

  4) 업데이트는 동시에 일어나지 않음

4.3 Implementation details

▼ 파라미터 설정(word2vec 기본 설정, 우리 모델에도 잘 맞음)

- 단어 벡터: dimension 300
- (5 랜덤 negative) / (positive 예시)
- 1 ≤ c = context windwo size ≤ 5
- rejection threshold = 10^{-4}
- dictionary는 트레이닝 셋에 최소 5번 이상 나온 단어들
- γ_0 = 0.025 (skipgram) or 0.05 (cbow)

- n-gram 사용한 우리 모델(C++베이스, 105k)? skipgram(145k)보다 학습이 1.5배 느림

4.4 Datasets

- 위키피디아 자료(9개 언어)
- Matt Mahoney의 전처리 스크립트 사용하여 정규화
- 무작위로 섞인 데이터셋 사용, 5번 패스해서 학습시킴

5. Results

아래 다섯가지 실험을 통해 모델을 평가함.

5.1 Human similarity judgement

- Word similarity dataset을 가지고 human judgement와 cosine similarity의 상관계수를 구함으로써 평가함.
- `sisg` (Subword Information Skip Gram) = n-gram으로 subword information을 사용한 model로, 본 논문에서 제안하는 모델임.
- 이 외 `cbow` 와 `skipgram`, `sisg-` 모델은 Training data에 없는 단어의 경우 out-of-Vaca 문제가 발생하기 때문에 null vector로 처리함.

결과 (Table)

		sg	cbow	sisg-	sisg
AR	WS353	51	52	54	**55**
DE	GUR350	61	62	64	**70**
	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**

Table 1.

- `sisg`가 English WS353을 제외한 모든 데이터에 대해 우수한 성능을 보임 .

→ WS353 dataset은 공통된 단어(Common words)로 구성되어 있기 때문. Rare Words dataset 에서는 우수한 성능을 보임.

- 모든 Dataset에 대해 `sisg`가 `sisg-` 보다 같거나 우수한 성능을 보임.

  → subword information을 제공하는 것의 장점 입증

- 영어보다는 Arabic / German / Russian data에 대해 좋은 성능을 보임.

5.2 Word analogy tasks

- 유추 질문 ⇒ *A is to B as C is to D*, 형태의 문장에서 model이 D를 예측하는 문제.

결과 Table

		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

Table 2.

- syntatic task 에 있어서는 baseline보다 우수한 성능을 보임.
- 그러나, semantic question에서는 좋지 않은 성적을 내고 심지어 German과 Italian에서는 성능이 떨어짐.

  → n-grams의 size가 적절히 선택되면 개선할 수 있음.

- 예상했듯이 Czech와 German같이 형태학적으로 풍부한 언어(morphologically rich language)에서 성능 개선이 눈에 띄게 관찰됨.

5.3 Comparison with morphological representations

단어 유사성 연구에서 subword information을 단어 벡터에 포함하는 다른 모델과의 비교

- 동일한 dataset을 학습한 후, `RNN` , `cbow` , `morphological transformation` 과 `sisg` 를 비교
  - `RNN` of Luong et al. (2013)
  - `cbow` of Qiu et al. (2014)
  - `morphological transformation` of Soricut and Och (2015) : 접두사와 접미사 분석(prefix and suffix analysis)에 기반한 모델

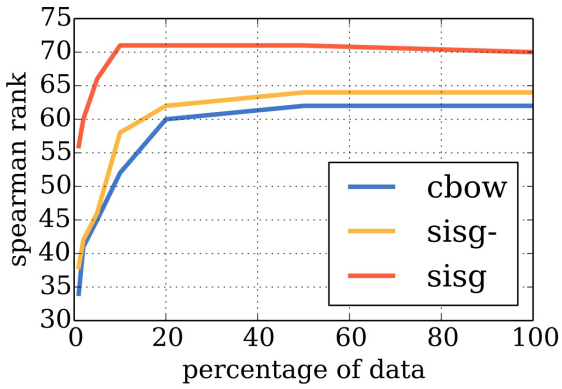
결과 Table

	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

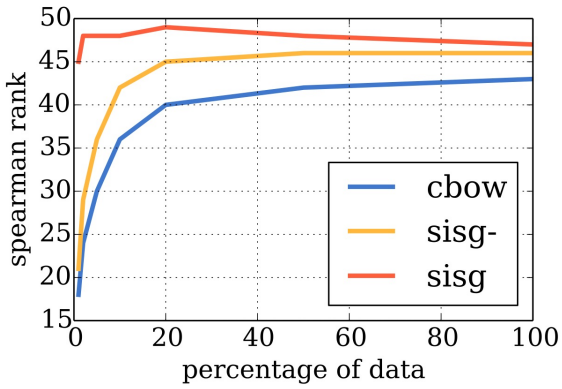
Table 3.

- 형태학적 분할자(morphological segmentors)에 기반한 모델보다 `sisg`가 더 좋은 성능을 냄.
- 특히, 명사 합성(noun compounding)을 모델링하지 않았기 때문에 German(DE)에서 큰 성능 차이가 남.

5.4 Effect of the size of the training data



(a) DE-GUR350



(b) EN-RW

단어 간의 문자 수준 유사성을 활용하기 때문에 자주 사용하지 않는 단어도 잘 모델링할 수 있음. 따라서 사용하는 훈련 데이터의 크기에도 더욱 견고해야함.

이를 평가하기 위해 위키백과 말뭉치에서 전체 데이터의 1, 2, 5, 10, 20 및 50%만큼 학습하여 spearman 상관계수로 나타냄.

데이터 세트가 축소됨에 따라 어휘 부족률이 증가하고 있으므로 sisg- 및 cbow의 성능은 저하
sisg은 이전에 보이지 않았던 단어에 사소한 벡터를 할당

1. 모든 데이터 세트와 모든 크기에서 sisg가 기준선보다 더 나은 성능을 보임. 그러나 점점 더 많은 데이터를 사용할 수록 cbow 모델의 성능은 향상되지만 sisg는 빠르게 포화 상태가 됨. 즉, 더 많은 데이터를 추가한다고 결과가 개선되진 않음.
2. sisg가 매우 작은 train dataset을 사용할 경우에도 좋은 단어 벡터를 제공한다.

5.5 Effect of the size of n-grams

	2	3	4	5	6
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

Syntatic task에서는 n을 작게 할 때 성능이 좋고
Semantic task에서는 n을 크게 할 때 성능이 좋다.

합성어가 많은 독일어 같은 경우 larger n-grams를 사용했을 때 성능이 높음

2-gram은 하나의 적절한 문자와 하나의 위치적인 문자로 구성되기 때문에, 결합이나 축약에 해당하는 접미사를 적절하게 캡처하기에 충분하지 않아 어떤 경우보다 성능이 떨어짐.

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**

sisg가 모든 언어에서 다른 모델(LSTM, sg)보다 좋은 성능을 보이는 것을 볼 수 있음.

사전 훈련된 단어 표현으로 언어 모델의 lookup table을 초기화하면 기준 LSTM에 대해 test perplexity가 개선된다.

특히 형태학적으로 풍부한 슬라브어족 언어(Cs(체코어), Ru(러시아어))에서 perplexity가 더 작은 것을 볼 수 있음

→ 언어 모델링 태스크에 subword 정보의 중요성을 보여주고 형태학적으로 풍부한 언어에 이 논문의 모델의 벡터가 유용하다는 것을 보여준다.

6. Qualitative analysis

6.1 Nearest neighbors.

query	tiling	tech-rich	english-born	micromanaging	eateries	dendritic
sisg	tile flooring	tech-dominated tech-heavy	british-born polish-born	micromanage micromanaged	restaurants eaterie	dendrite dendrites
sg	bookcases built-ins	technology-heavy .ixic	most-capped ex-scotland	defang internalise	restaurants delis	epithelial p53

코사인 유사도를 사용하여 주어진 단어와 가장 유사한 단어 2개를 뽑은 테이블.

sisg(subword information + skip-gram)가 sg(skip-gram)보다 구조(형태)적으로 유사한 단어를 더 찾아줌.

tech-rich라는 단어를 봤을 때 → tech-dominated, tech-heavy(sisg)이고 / technology-heavy, .ixic(sg) 의미적으로는 sg가 유사할 수 있지만 구조(형태)적으로 봤을 때 sisg가 매우 유사하다.

6.2 Character n-grams and morphemes

	word	n-grams		
DE	autofahrer	fahr	fahrer	auto
	freundeskreis	kreis	kreis>	<freun
	grundwort	wort	wort>	grund
	sprachschule	schul	hschul	sprach
	tageslicht	licht	gesl	tages
EN	anarchy	chy	<anar	narchy
	monarchy	monarc	chy	<monar
	kindness	ness>	ness	kind
	politeness	polite	ness>	eness>
	unlucky	<un	cky>	nlucky
	lifetime	life	<life	time
	starfish	fish	fish>	star
	submarine	marine	sub	marin
	transform	trans	<trans	form
FR	finirais	ais>	nir	fini
	finissent	ent>	finiss	<finis
	finissions	ions>	finiss	sions>

주어진 단어에서 가장 중요한 3개의 character n-gram을 추출한 결과

→ 합성 명사와 접두사 접미사를 잘 표현함

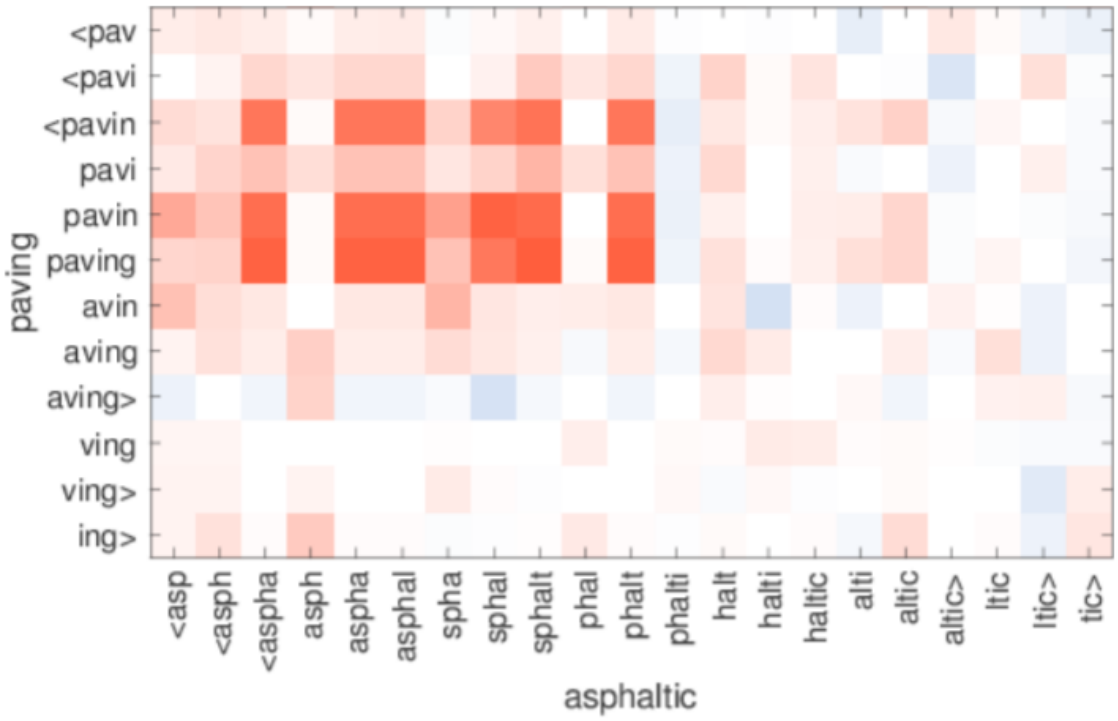
ex - autofahrer → auto + fahrer

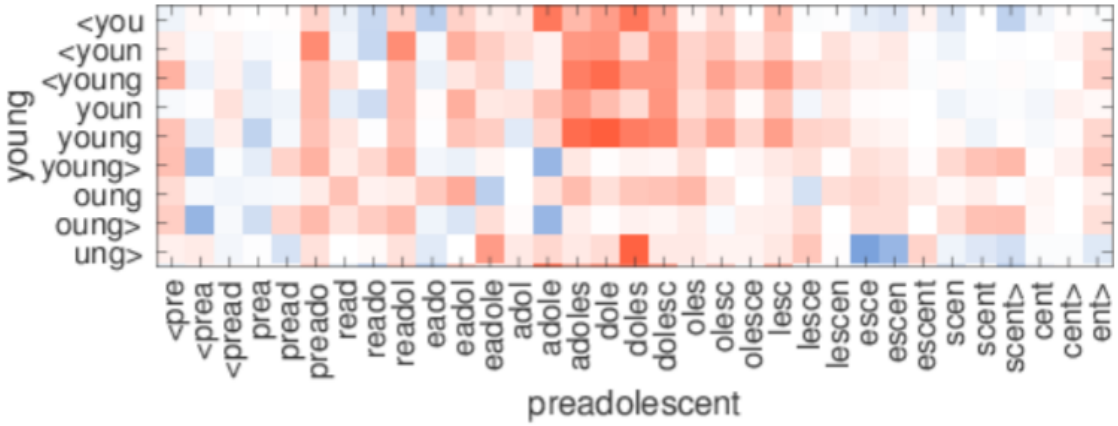
kindness → kind + ness>

starfish → star + fish

결과를 봤을 때, 중요한 단어들이 n-grams에 다 표현됨.

6.3 Word similarity for OOV words





X축 : OOV(등장하지 않은 단어)

Y축 : 학습 데이터 셋 내 단어

X, Y축 사이의 character n-gram 유사도 → 비슷한 뜻을 가진 단어들끼리 유사도가 높은 것(짙은 빨강)을 볼 수 있다.

"pav-", "-sphal-" : (아스팔트) 포장

"young", "adoles" : 어린(이)

7. Conclusion

기존 skip-gram 기반의 단어 표현에 character n-gram으로 subword information을 넣어봤을 때, 성능이 더 좋고 형태(구조)적 분석에 의존한 방법임을 알 수 있다.