Enriching Word Vectors with Subword Information

2018.12.08
김보섭

Agenda

1. **Abstract**
2. **Introduction**
3. **Model**
4. **Experimental setup**
5. **Results**
6. **Qualitative analysis**
7. **Conclusion**

Abstract

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

Abstract

Continuous word representations, trained on large unlabeled corpora are useful for many natural language processing tasks. Popular models that learn such representations ignore the morphology of words, by assigning a distinct vector to each word. This is a limitation, especially for languages with large vocabularies and many rare words. In this paper, we propose a new approach based on the skipgram model, where each word is represented as a bag of character n-grams. A vector representation is associated to each character n-gram; words being represented as the sum of these representations. Our method is fast, allowing to train models on large corpora quickly and allows us to compute word representations for words that did not appear in the training data. We evaluate our word representations on nine different languages, both on word similarity and analogy tasks. By comparing to recently proposed morphological word representations, we show that our vectors achieve state-of-the-art performance on these tasks.

Introduction (1/3)

대부분의 word의 continuous representation (aka. word vector)을 만들어내는 방법론은 parameter sharing을 하지않음

Count based vs direct prediction

- LSA, HAL (Lund & Burgess),
- COALS, Hellinger-PCA (Rohde et al, Lebret & Collobert)

- Fast training
- Efficient usage of statistics
- Primarily used to capture word similarity
- Disproportionate importance given to large counts

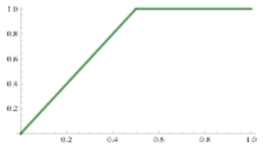
- Skip-gram/CBOW (Mikolov et al)
- NNLM, HLBL, RNN (Bengio et al; Collobert & Weston; Huang et al; Mnih & Hinton)

- Scale with corpus size
- Inefficient usage of statistics
- Generate improved performance on other tasks
- Can capture complex patterns beyond word similarity

Combining the best of both worlds: GloVe

$$J(\theta) = \frac{1}{2}\sum_{i,j=1}^{W} f(P_{ij})(u_i^T v_j - \log P_{ij})^2$$

- Fast training
- Scalable to huge corpora

$f \sim$ 

- Good performance even with small corpus, and small vectors
- By Pennington, Socher, Manning (2014)

위와 같은 방법론들은 word의 internal structure 고려하지 않음
→ morphologically rich language (eg. French, Spanish) 적용하기 힘듦

Stanford CS 224N course (http://web.stanford.edu/class/cs224n/lectures/lecture3.pdf)

Introduction (2/3)

character-level information을 이용하면 word의 internal structure을 고려
→ Out of Vocabulary (OOV)대처, 더 좋은 representation을 얻을 수 있음

Ras+po+*lag*+a+ušč+ej

Disposing (inpraes,dat,sg,partcp,plen,f,ipf,intr)

OSM CNN_{char}	OSM $BILSTM_{char}$
ras+po+lag+a+ušč+iy *disposing (inpraes,nom,sg,partcp,plen,m,ipf,inan,intr)*	ras+slab+l+ja+ušč+ej *relaxing (inpraes,dat,sg,partcp,plen,f,ipf)*
ras+po+lag+a+ušč+im *disposing (inpraes,ins,sg,partcp,plen,m,ipf,intrn)*	so+pro+voj+d+a+ušč+ej *accompanying (inpraes,dat,sg,partcp,plen,f,ipf,tran)*
ras+po+lag+a+ušč+ie *disposing (inpraes,nom,pl,partcp,plen,ipf,intr)*	ras+slab+l+ja+ušč+uju *relaxing (inpraes,acc,sg,partcp,plen,f,ipf)*
ras+po+lag+a+ušč+ih *disposing (inpraes,gen,pl,partcp,plen,ipf,intr)*	ras+po+lag+a+ušč+iy *disposing (inpraes,nom,sg,partcp,plen,m,ipf,inan,intr)*
ras+po+lag+a+ušč+i+e+sja *disposing (inpraes,nom,pl,partcp,plen,ipf,act)*	pro+dvig+a+ušč+ej *promoting (inpraes,dat,sg,partcp,plen,f,ipf,act)*

S+*konfigur*+ir+ova+ť

Configure (v,pf,tran,inf)

OSM CNN_{char}	OSM $BILSTM_{char}$
s+konfigur+ir+ui+te *configure (v,pf,tran,pl,imper,2p)*	konfigur+ir+ova+ť *configure (v,ipf,tran,inf)*
s+konfigur+ova+li *configured (v,pf,tran,praet,pl,indic)*	s+korrekt+ir+ova+ť *adjust (v,pf,tran,inf)*
s+konfigur+ova+n *configured (v,pf,tran,praet,sg,partcp,brev,m,pass)*	s+*koordin*+ir+ova+ť *coordinate (v,pf,tran,inf)*
s+konstru+ir+ova+ť *construct (v,pf,tran,inf)*	s+fokus+ir+ova+ť *focus (v,pf,tran,in)*
s+kompil+ir+ova+ť *compile (v,pf,tran,inf)*	s+kompil+ir+ova+ť *compile (v,pf,tran,inf)*

Table 5: Analysis of the five most similar Russian words (initial word is OOV), under the OSM CNN_{char} and OSM $BILSTM_{char}$ word encodings based on cosine similarity. The diacritic ´ indicates softness. **POS tags**: *s*-noun, *a*-adjective, *v*-verb; **Gender**: *m*-masculine, *f*-feminine, *n*-neuter; **Number**: *sg*-singular, *pl*-plural; **Case**: *nom*-nominative, *gen*-genitive, *dat*-dative, *acc*-accusative, *ins*-instrumental, *abl*-prepositional, *loc*-locative; **Tense**: *praes*-present, *inpraes*-continuous, *praet*-past, *pf*-perfect, *ipf*-imperfect; *indic*-indicative; **Transitivity**: *trans*-transitive, *intr*-intransitive; **Adjective form**: *br*-brevity, *plen*-full form, *poss*-possessive; **Comparative**: *supr*-superlative, *comp*-comparative; **Noun person**: *1p*-first, *2p*-second, *3p*-third;

Vylomova, Ekaterina, et al. "Word representation models for morphologically rich languages in neural machine translation." arXiv preprint arXiv:1606.04217 (2016)

Introduction (3/3)

본 논문에서는 character-level information을 고려 → subword information을 고려하는 skip-gram 기반의 모형을 제안 (aka, FastText)

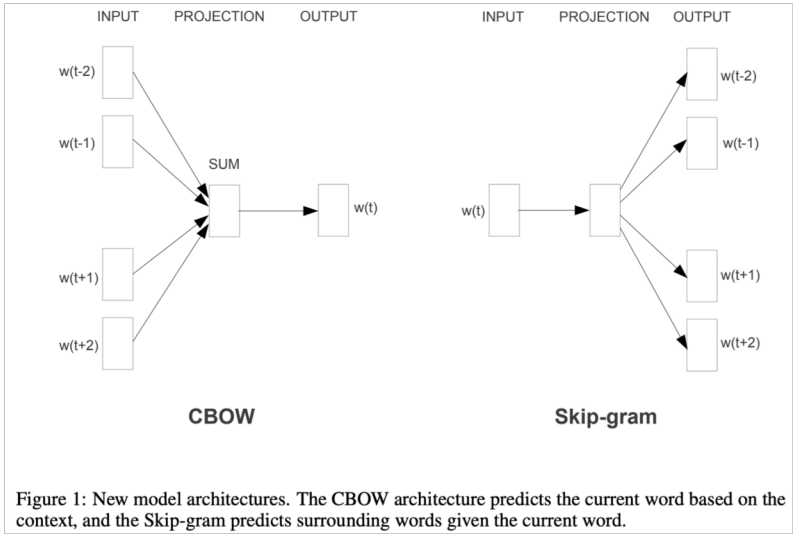


Figure 1: New model architectures. The CBOW architecture predicts the current word based on the context, and the Skip-gram predicts surrounding words given the current word.

Subword information

eg. Where
→ <wh, whe, her, ere, re>, <where>

In this paper, we propose to learn representations for character n-grams, and to represent words as the sum of the n-gram vectors. Our main contribution is to introduce an extension of the continuous skip-gram model (Mikolov et al., 2013b), which takes into account subword information. We evaluate this model on nine languages exhibiting different morphologies, showing the benefit of our approach.

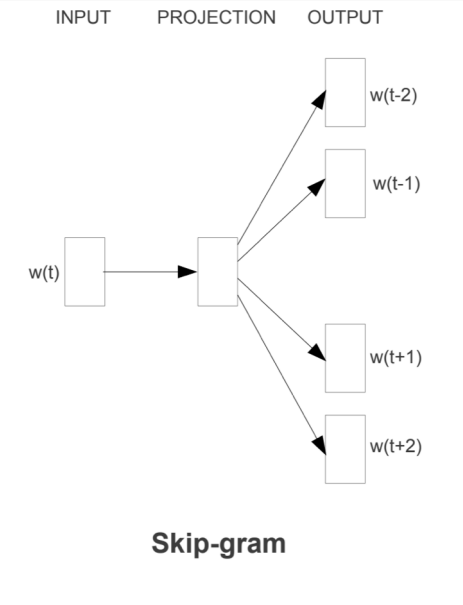
Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).

Model (1/2)

skip-gram을 기반으로하여 morphology을 고려하기위해, subword (character n-gram)의 representation의 합으로 word representation을 표현

score function을 subword information을 고려하고, parameter sharing을 할 수 있도록 변경

$$s(w_t, w_c) : u_{w_t}^T v_{w_c} \rightarrow \sum_{g \in G_{w_t}} z_g^T v_c$$



$$u_{w_t} = Uw_t, v_{w_{t-2}} = V_{W_{t-2}},$$

$\dim(U) = (d, V), \dim(V) = (V, d)$
$u_{w_t} \in R^d, v_{w_{t-2}} \in R^d$

The problem of predicting context words can instead be framed as a set of independent binary classification tasks. Then the goal is to independently predict the presence (or absence) of context words. For the word at position t we consider all context words as positive examples and sample negatives at random from the dictionary. For a chosen context position c, using the binary logistic loss, we obtain the following negative log-likelihood:

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

where $\mathcal{N}_{t,c}$ is a set of negative examples sampled from the vocabulary. By denoting the logistic loss function $\ell : x \mapsto \log(1 + e^{-x})$, we can re-write the objective as:

$$\sum_{t=1}^{T} \left[\sum_{c \in \mathcal{C}_t} \ell(s(w_t,\, w_c)) + \sum_{n \in \mathcal{N}_{t,c}} \ell(-s(w_t,\, n)) \right].$$

A natural parameterization for the scoring function s between a word w_t and a context word w_c is to use word vectors. Let us define for each word w in the vocabulary two vectors u_w and v_w in $\mathbb{R}^d$. These two

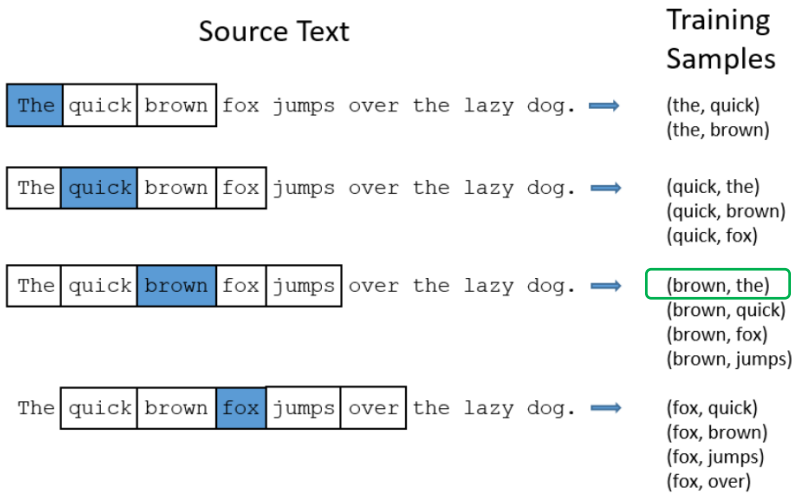
Suppose that you are given a dictionary of n-grams of size G. Given a word w, let us denote by $\mathcal{G}_w \subset \{1, \ldots, G\}$ the set of n-grams appearing in w. We associate a vector representation $\mathbf{z}_g$ to each n-gram g. We represent a word by the sum of the vector representations of its n-grams. We thus obtain the scoring function:

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

This simple model allows sharing the representations across words, thus allowing to learn reliable representation for rare words.

Model (2/2)

skip-gram을 기반으로 하여 morphology을 고려하기 위해, subword (character n-gram)의 representation의 합으로 word representation을 표현



character 3-grams case

$$s(w_{brown}, w_{the}) = \sum_{g \in G_{w_{brown}}} z_g^T v_{w_{the}}$$

$$u_{w_{brown}} = \sum_{g \in G_{w_{brown}}} z_g$$

$G_{w_{brown}} \in \boldsymbol{G}\ (char\ n-grams)$
$G_{w_{brown}} = \{< br, bro, row, own, wn >, < brown >\}$
$z_g \in R^d, v_{w_{the}} \in R^d$

Experimental setup

skip-gram, cbow (continuous bag of words) 방식과 비교, 학습하는 word vector는 모두 300 dimension, negative sampling 방식으로 학습

4.2 Optimization

We solve our optimization problem by performing stochastic gradient descent on the negative log likelihood presented before. As in the baseline `skipgram` model, we use a linear decay of the step size. Given a training set containing T words and a number of passes over the data equal to P, the step size at time t is equal to $\gamma_0(1 - \frac{t}{TP})$, where γ_0 is a fixed parameter. We carry out the optimization in parallel, by resorting to Hogwild (Recht et al., 2011). All threads share parameters and update vectors in an asynchronous manner.

4.3 Implementation details

For both our model and the baseline experiments, we use the following parameters: the word vectors have dimension 300. For each positive example, we sample 5 negatives at random, with probability proportional to the square root of the uni-gram frequency. We use a context window of size c, and uniformly sample the size c between 1 and 5. In order to subsample the most frequent words, we use a rejection threshold of 10^{-4} (for more details, see (Mikolov et al., 2013b)). When building the word dictionary, we keep the words that appear at least 5 times in the training set. The step size γ_0 is set to 0.025 for the `skipgram` baseline and to 0.05 for both our model and the `cbow` baseline. These are the default values in the `word2vec` package and work well for our model too.

Using this setting on English data, our model with character n-grams is approximately $1.5\times$ slower to train than the `skipgram` baseline. Indeed, we process 105k words/second/thread versus 145k words/second/thread for the baseline. Our model is implemented in C++, and is publicly available.[3]

Results (1/5)

Human similarity judgement와의 비교를 통해서, 제안한 방법이 OOV, Rare words, morphologically rich language에 대해서 성능이 좋음을 보임

		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: Correlation between human judgement and similarity scores on word similarity datasets. We train both our model and the `word2vec` baseline on normalized Wikipedia dumps. Evaluation datasets contain words that are not part of the training set, so we represent them using null vectors (`sisg-`). With our model, we also compute vectors for unseen words by summing the n-gram vectors (`sisg`).

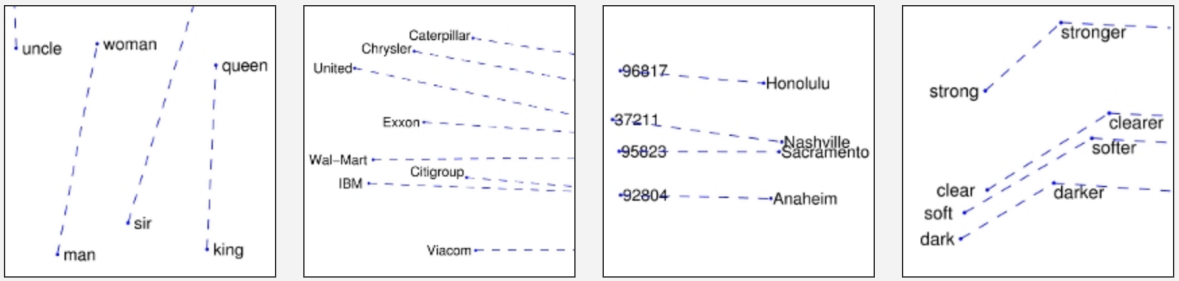
Results (2/5)

Word analogy tasks에서 특히 제안한 방법론이 syntactic tasks에서 좋은 성능을 보임을 알 수 있음

2. Linear substructures

The similarity metrics used for nearest neighbor evaluations produce a single scalar that quantifies the relatedness of two words. This simplicity can be problematic since two given words almost always exhibit more intricate relationships than can be captured by a single number. For example, *man* may be regarded as similar to *woman* in that both words describe human beings; on the other hand, the two words are often considered opposites since they highlight a primary axis along which humans differ from one another.

In order to capture in a quantitative way the nuance necessary to distinguish *man* from *woman*, it is necessary for a model to associate more than a single number to the word pair. A natural and simple candidate for an enlarged set of discriminative numbers is the vector difference between the two word vectors. GloVe is designed in order that such vector differences capture as much as possible the meaning specified by the juxtaposition of two words.



man - woman | company - ceo | city - zip code | comparative - superlative

The underlying concept that distinguishes *man* from *woman*, i.e. sex or gender, may be equivalently specified by various other word pairs, such as *king* and *queen* or *brother* and *sister*. To state this observation mathematically, we might expect that the vector differences *man* - *woman*, *king* - *queen*, and *brother* - *sister* might all be roughly equal. This property and other interesting patterns can be observed in the above set of visualizations.

		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: Accuracy of our model and baselines on word analogy tasks for Czech, German, English and Italian. We report results for semantic and syntactic analogies separately.

https://nlp.stanford.edu/projects/glove/

Results (3/5)

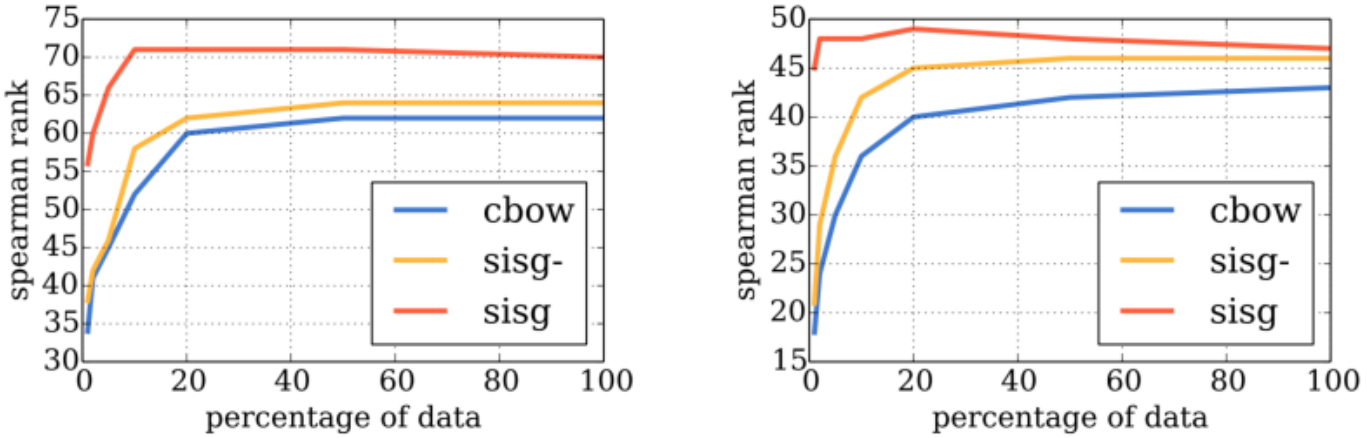
character n-gram만 사용해도, Morphological feature engineering하여 활용하는 다른 연구들보다 더 좋은 성능을 보임을 알 수 있음

	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: Spearman's rank correlation coefficient between human judgement and model scores for different methods using morphology to learn word representations. We keep all the word pairs of the evaluation set and obtain representations for out-of-vocabulary words with our model by summing the vectors of character n-grams. Our model was trained on the same datasets as the methods we are comparing to (hence the two lines of results for our approach).

Results (4/5)

OOV에 대처할 수 있고, internal structure를 고려하므로 training corpus가 적어도 상대적으로 다른 방법론에 비해서 잘 학습함을 알 수 있음



(a) DE-GUR350 (b) EN-RW

Figure 1: Influence of size of the training data on performance. We compute word vectors following the proposed model using datasets of increasing size. In this experiment, we train models on a fraction of the full Wikipedia dump.

Results (5/5)

character n-gram에서 n >=5 이상 활용하는 것이 좋으며, SISG 기반의 word vector를 사용하면 Language Modeling 성능도 좋아짐

	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

Table 4: Study of the effect of sizes of n-grams considered on performance. We compute word vectors by using character n-grams with n in $\{i, \ldots, j\}$ and report performance for various values of i and j. We evaluate this effect on German and English, and represent out-of-vocabulary words using subword information.

	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**

Table 5: Test perplexity on the language modeling task, for 5 different languages. We compare to two state of the art approaches: CLBL refers to the work of Botha and Blunsom (2014) and CANLM refers to the work of Kim et al. (2016).

Qualitative analysis (1/2)

character n-gram 방식은 word의 중요한 morpheme을 어느 정도 modeling하며, OOV에 대한 word similarity도 잘 modeling 할 수 있음

6.1 Nearest neighbors.

We report sample qualitative results in Table 7. For selected words, we show nearest neighbors according to cosine similarity for vectors trained using the proposed approach and for the `skipgram` baseline. As expected, the nearest neighbors for complex, technical and infrequent words using our approach are better than the ones obtained using the baseline model.

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

Table 7: Nearest neighbors of rare words using our representations and `skipgram`. These hand picked examples are for illustration.

6.2 Character n-grams and morphemes

We want to qualitatively evaluate whether or not the most important n-grams in a word correspond to morphemes. To this end, we take a word vector that we construct as the sum of n-grams. As described in Sec. 3.2, each word w is represented as the sum of its n-grams: $u_w = \sum_{g \in \mathcal{G}_w} z_g$. For each n-gram g, we propose to compute the restricted representation $u_{w \backslash g}$ obtained by omitting g:

$$u_{w \backslash g} = \sum_{g' \in \mathcal{G} - \{g\}} z_{g'}.$$

We then rank n-grams by increasing value of cosine between u_w and $u_{w \backslash g}$. We show ranked n-grams for selected words in three languages in Table 6.

	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>

Table 6: Illustration of most important character n-grams for selected words in three languages. For each word, we show the n-grams that, when removed, result in the most different representation.

Qualitative analysis (2/2)

character n-gram 방식은 word의 중요한 morpheme을 어느 정도 modeling하며, OOV에 대한 word similarity도 잘 modeling 할 수 있음

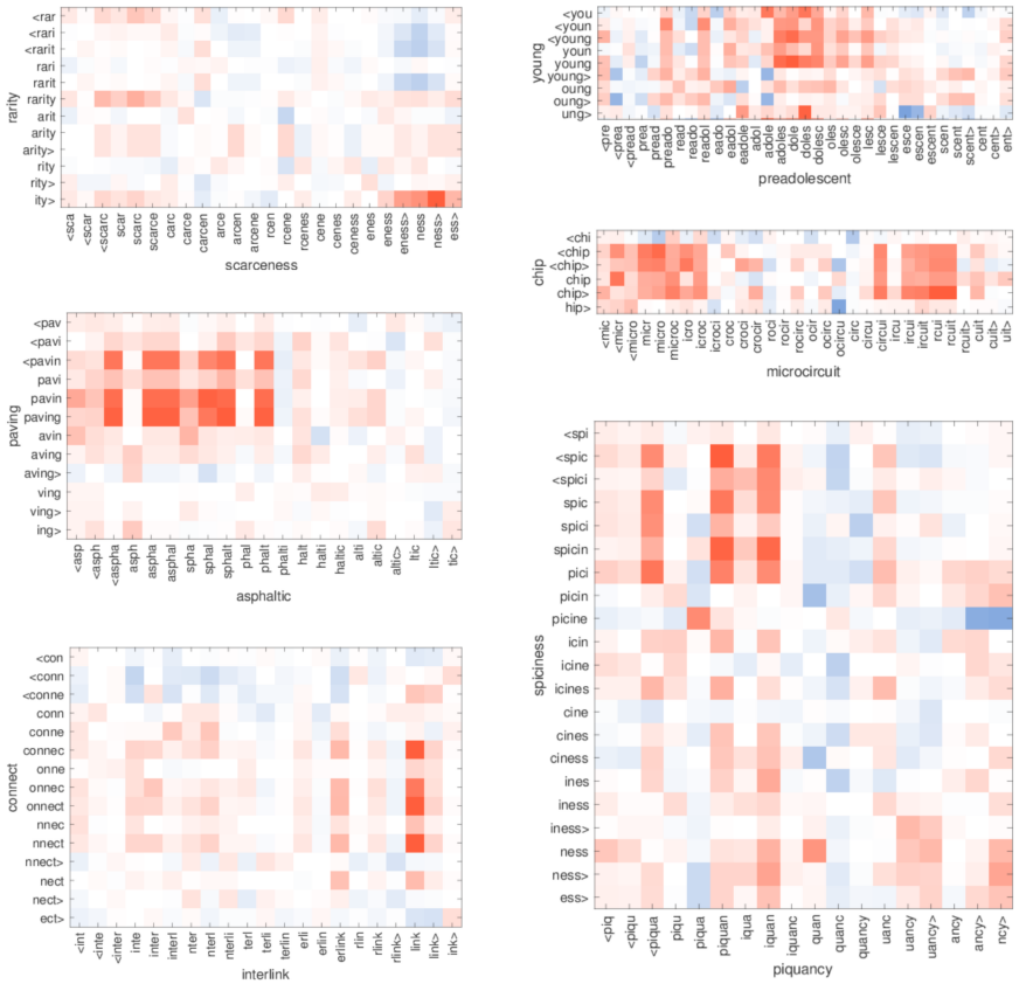


Figure 2: Illustration of the similarity between character n-grams in out-of-vocabulary words. For each pair, only one word is OOV, and is shown on the x axis. Red indicates positive cosine, while blue negative.

Conclusion

7 Conclusion

In this paper, we investigate a simple method to learn word representations by taking into account subword information. Our approach, which incorporates character n-grams into the skipgram model, is related to an idea that was introduced by Schütze (1993). Because of its simplicity, our model trains fast and does not require any preprocessing or supervision. We show that our model outperforms baselines that do not take into account subword information, as well as methods relying on morphological analysis. We will open source the implementation of our model, in order to facilitate comparison of future work on learning subword representations.

Q & A

감사합니다.