FastText Paper Review

(Enriching Word Vectors with Subword Information, 2017)

집현전 초급반 이기창
(efrks@naver.com)

Index

1. Author
2. Abstract
3. Introduction
4. Subword Model
5. Results
6. Conclusion

Author

P.Bojanowski

Facebook AI Researcher (2016~)

Paper

Enriching Word Vectors with Subword Information (2017)
Bag of Tricks for Efficient Text Classification (2017)
Learning Word Vectors for 157 Languages (2018)

Author

T.Mikolov

Facebook AI Researcher (2016~2020)

CIIRC CTU Prague Researcher(2020~)

Paper

Distributed representations of words and phrases and their compositionality (2013a)
Efficient Estimation of Word Representations in Vector Space (2013b)

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, allow-

기존 모델은 **단어마다 다른 벡터를 할당하여 단어의 형태를 무시**하게 된다

논문에서는 Skip-gram을 기반으로 한 모델에 **각 단어를 철자(Character) n-gram 벡터의 조합**으로 표현하여 이를 해결하였다

Abstract

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.

학습 속도가 빠르고,

학습 데이터에 등장하지 않은 단어도 표현 가능

9개 언어에 대해

단어 유사도 및 추론 태스크를 통해 평가

→ **SOTA 달성**

Introduction

Most of these techniques represent each word of the vocabulary by a distinct vector, without parameter sharing. In particular, they ignore the internal structure of words, which is an important limitation for morphologically rich languages, such as Turkish or Finnish. For example, in French or Spanish, most verbs have more than forty different inflected forms, while the Finnish language has fifteen cases for nouns. These languages contain many word forms that occur rarely (or not at all) in the training corpus, making it difficult to learn good word representations. Because many word formations follow rules, it is possible to improve vector representations for morphologically rich languages by using character level information.

기존 모델은 다른(=parameter를 공유하지 않는) 벡터로 단어를 표현하였다.

형태학적으로 복잡한 언어는 잘 표현하지 못함

Ex) Turkish, Finnish

단어의 형태가 일정한 룰을 따르고 있으니

철자 단위 정보를 사용하여

더 좋은 단어 표현을 만들자

Introduction

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.

기존 skip-gram으로 학습한 벡터에
철자 단위 **n-gram 벡터의 합을 더해주어**
단어 표현을 풍부하게 만들었음

Subword Model

3.2 Subword model

By using a distinct vector representation for each word, the skipgram model ignores the internal structure of words. In this section, we propose a different scoring function s, in order to take into account this information.

Each word w is represented as a bag of character n-gram. We add special boundary symbols < and > at the beginning and end of words, allowing to distinguish prefixes and suffixes from other character sequences. We also include the word w itself in the set of its n-grams, to learn a representation for each word (in addition to character n-grams). Taking the

위처럼 단어의 양 끝에 <,> 를 더하여

접두사와 접미사를 구분할 수 있도록 했다

이미지 출처 : https://amitness.com/2020/06/fasttext-embeddings/

Subword Model

<wh, whe, her, ere, re>

and the special sequence

<where>.

Note that the sequence <her>, corresponding to the word *her* is different from the tri-gram her from the word *where*. In practice, we extract all the n-grams for n greater or equal to 3 and smaller or equal to 6. This is a very simple approach, and different sets of n-grams could be considered, for example taking all prefixes and suffixes.

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>

3 ≤ n ≤ 6 범위의 n-gram을 사용하였다

이미지 출처 : https://amitness.com/2020/06/fasttext-embeddings/

Subword Model

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.

In order to bound the memory requirements of our model, we use a hashing function that maps n-grams to integers in 1 to K. We hash character sequences using the Fowler-Noll-Vo hashing function (specifically the `FNV-1a` variant).[1] We set $K = 2.10^6$ below. Ultimately, a word is represented by its index in the word dictionary and the set of hashed n-grams it contains.

단어를 **n-gram 벡터의 합**으로 나타냄

단어 간에 표현을 **공유**하도록 하여
희소 단어도 의미 있는 표현을 배움

Results – (1) word similarity

		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`).

Baseline model : sg, cbow

sisg-, sisg 와 비교

sisg- : OOV에 대해 **null vector**로 표현

sisg : **n-gram 벡터의 합**으로 표현

Arabic, German, Russian에서 더 좋은 성능

[**형태적으로 복잡**하거나

합성어(compound words)가 많은 언어]

WS353보다 RW(Rare words dataset)에서

더 좋은 성능을 보인다

Results – (2) word analogy

		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.

단어 추론과 관련된 태스크에서는

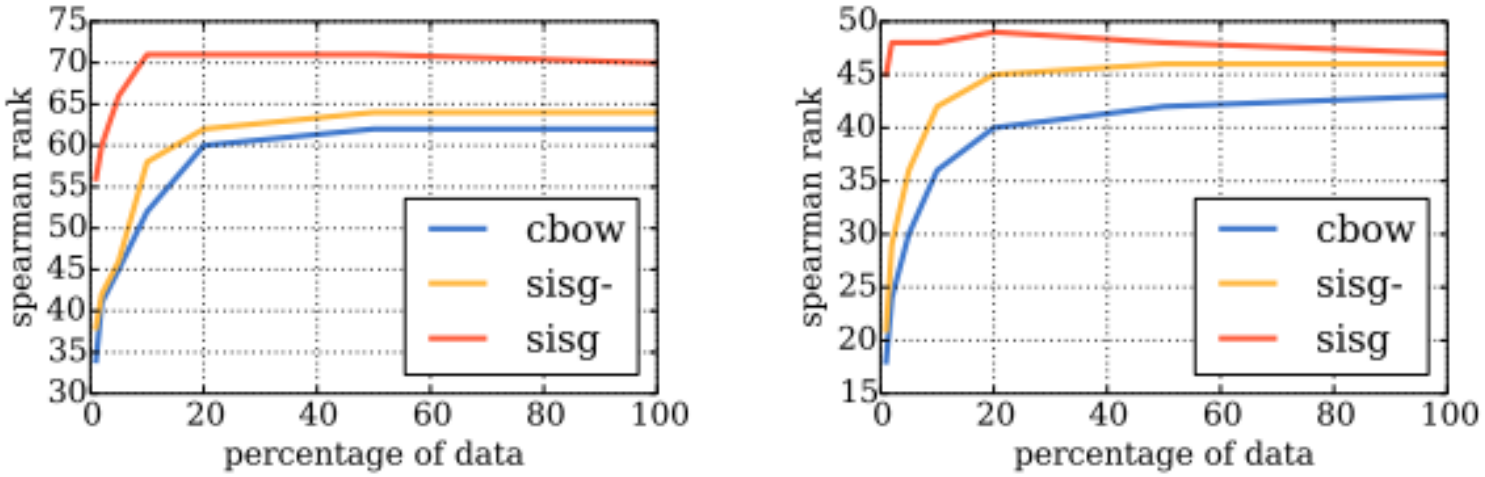
의미적(Semantic)인 추론보다

구조적(Syntactic)인 추론에서

기존 모델에 비해 뛰어난

성능 향상이 있다

Results – (3) data size



(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.

데이터가 **적을 때**

(전체 데이터의 1~20%)

확실히 이전 모델보다

좋은 성능을 보임

Results – (4) size of n-gram

	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.

Syntactic task 에서는

작은 n을 고려할 때 성능이 좋고

Semantic task 에서는

큰 n만 고려할 때 성능이 좋다

Results – (5) language model

	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).

모든 언어에서 다른 모델보다
더 좋은 성능을 보였으며

특히, **Vocab size**가 큰
슬라브 언어(Czech, Russian)에서
Perplexity의 감소폭이 더 두드러진다

Results - (6) Qualitative analysis : 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

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

코사인 유사도를 사용하여
주어진 단어와 가장 유사한 단어
2개를 뽑아낸 결과

Skip-gram만 사용했을 때보다
Subword information을 사용하면
구조적으로 유사한 단어를 잘 찾음

Results – (6) Qualitative analysis : important n-grams

	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.

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

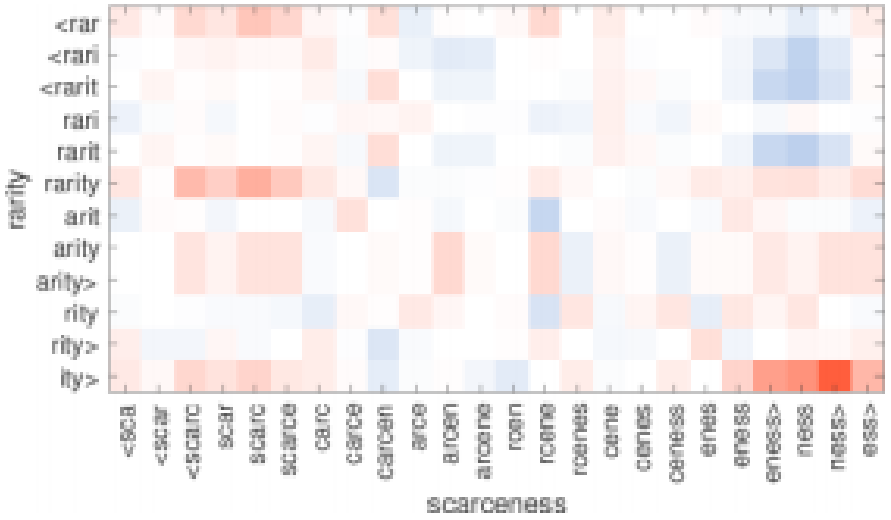
합성 명사와 **접두사, 접미사**를 잘 표현함

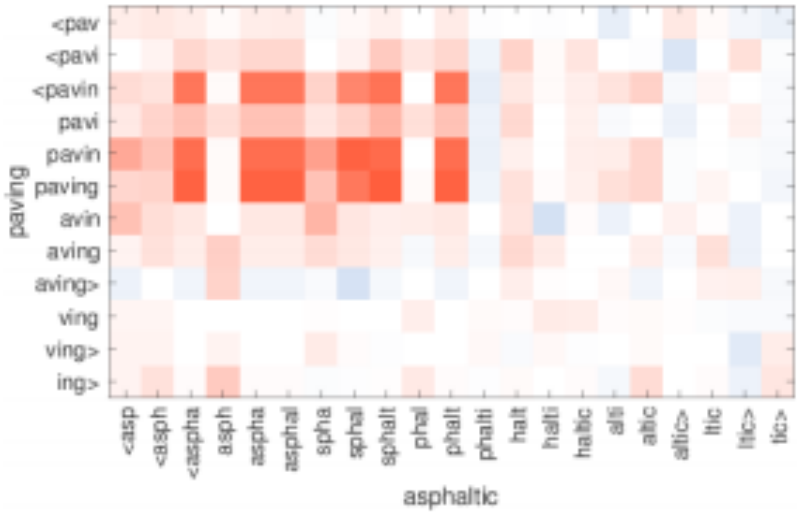
Ex) starfish = star + fish

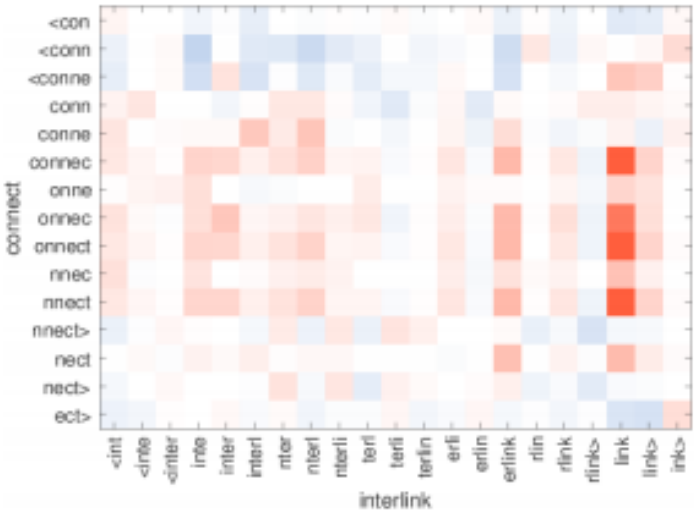
politeness = polite + ness〉

transform = 〈trans + form

Results - (6) Qualitative analysis : similarity for OOV







X축 : 등장하지 않은 단어(OOV)

Y축 : 학습 데이터셋 내의 단어(Y축)

사이의 character n-gram 유사도

“ity>” - “ness>” : 명사형 접미사

“pav-” - “-sphal-” : (아스팔트) 포장

“nnect, onnect” - “link” : 연결

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.

기존 skip-gram 기반의 단어 표현에
character n-gram으로
subword information을 고려하였다.

논문을 통해 제시한 방법이 성능이 더 좋고
형태적 분석에 의존한 방법임을 알 수 있었다.

시청해 주셔서 감사합니다 :)