Generalizing Word Embeddings using Bag of Subwords

Jinman Zhao, Sidharth Mudgal, Yingyu Liang
University of Wisconsin-Madison
Nov. 2, 2018 @ EMNLP



Word Embeddings

Belgium officially the Kingdom of Belgium, is a country in Western Europe bordered by France, the Netherlands, Germany and Luxembourg. It covers an area of 30,528 square kilometres (11,787 sq mi) and has a population of more than 11.4 million. The capital and largest city is Brussels; other major cities are Antwerp, Ghent, Charleroi and Liège. The sovereign state of Belgium is a federal constitutional monarchy with a parliamentary system of governance. Its institutional organisation is complex and is structured on both regional and linguistic grounds.

Text corpus

Train

the	→	[-0.1	0.1	0.3	...]
be	→	[0.2	-0.3	0.2	...]
and	→	[0.1	0.1	-0.1	...]
...					
Belgium	→	[0.3	-0.4	0.5	]
Brussels	→	[0.2	-0.3	0.6	]
...					
Belgian	→	[0.2	-0.6	0.4	...]
decomposable	→	[?	?	?	...]
preEMNLP	→	[?	?	?	...]

Model

Word Embedding and Vocabulary

Word embedding word $\mapsto$ word vector

Learnt from large text corpus.

Essential to many neural-network based approaches for NLP tasks.

Many popular word embedding techniques assume fixed-size vocabularies.

E.g. word2vec (Mikolov et al. , 2013), GloVe (Pennington et al. , 2014).

They have little to do with out-of-vocabulary (OOV) words!

Generalize to OOV words?

1. Estimating word vectors for rare or unseen words can be crucial.

    Understanding new trending terms.

2. We can often guess the meaning of the word from its spelling.

    “preEMNLP” probably means “before EMNLP”.
    +ese means the people of some place.
    Chemical names.

Generalize to OOV words?

1. Estimating word vectors for rare or unseen words can be crucial.

    Understanding new trending terms.

2. We can often guess the meaning of the word from its spelling.

    “preEMNLP” probably means “before EMNLP”.
    +ese means the people of some place.
    Chemical names.

0. Existence of good pre-trained vectors (with fixed-size vocabularies).

Our Approach: A Learning Task

Generalizes pre-trained word embeddings

$$\text{Vocabulary} \rightarrow R^n$$

$$\text{word} \mapsto \text{word vector}$$

towards OOV words by using them as training data and learning a mapping

$$\textbf{spelling} \mapsto \textbf{word vector}$$

No context is needed!

Our Bag-of-Subwords Model

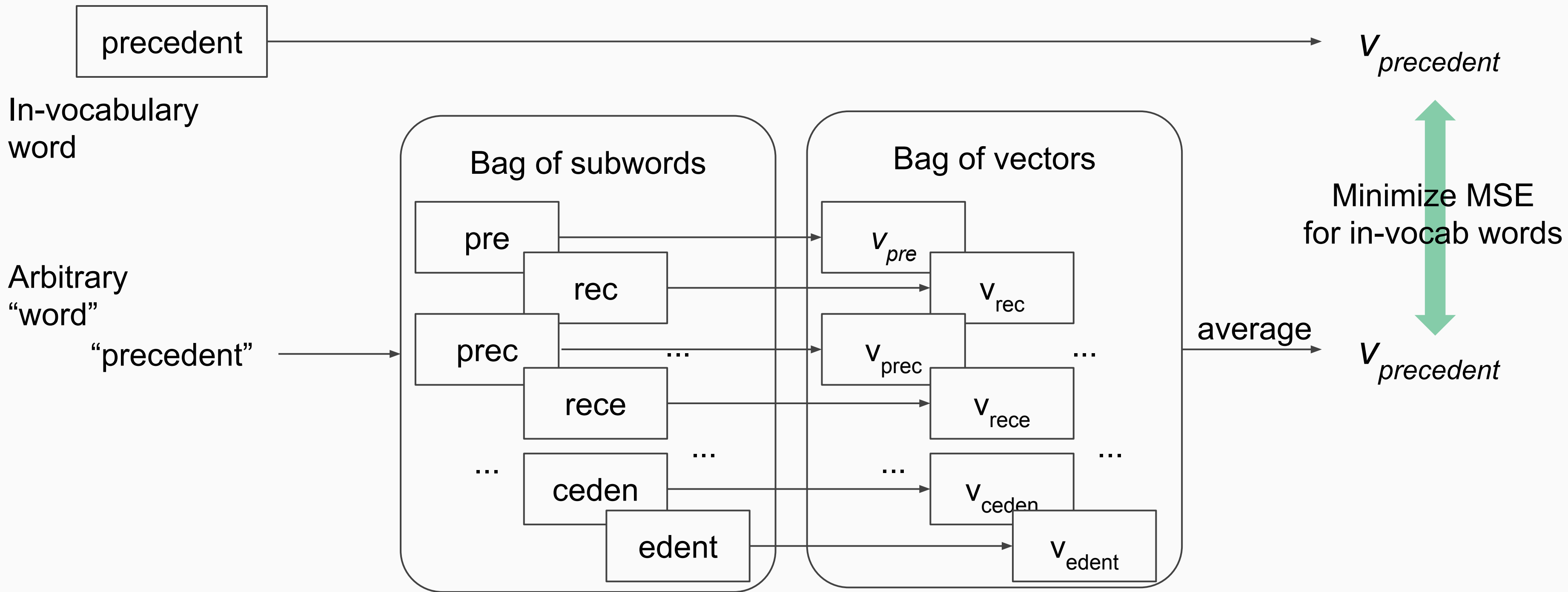
Parameters: a lookup table maps character n-grams to vectors.

Word vector = average of the vectors of all its character n-grams.

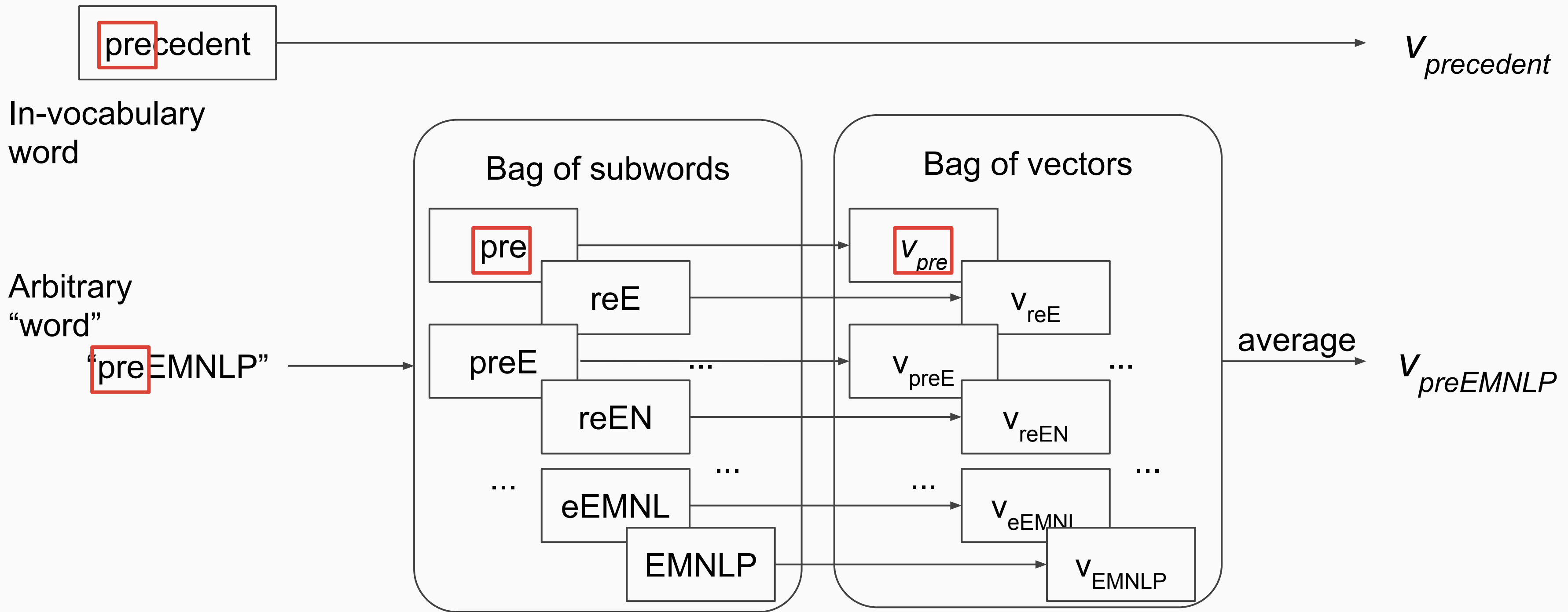
- Limit the sizes of character n-grams to be within l_{min} and l_{max}.

Training: minimize mean square loss between BoS vector and target vector for all words in the vocabulary.

Bag-of-Subwords Model



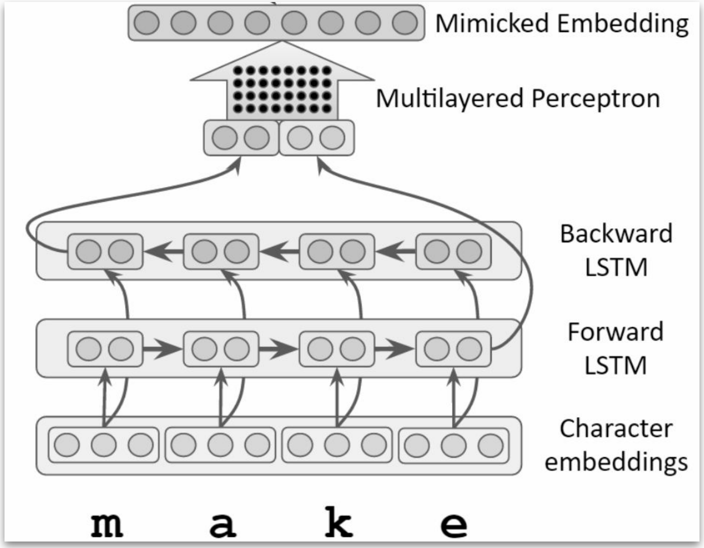
Bag-of-Subwords Model



Most Related Works

MIMICK (Pinter et al. 2017) tacles the same task using a character-level bidirectional LSTM model.

fastText (Bojanowski et al., 2017) uses the same subword-level character n-gram model but is trained over large text corpora.



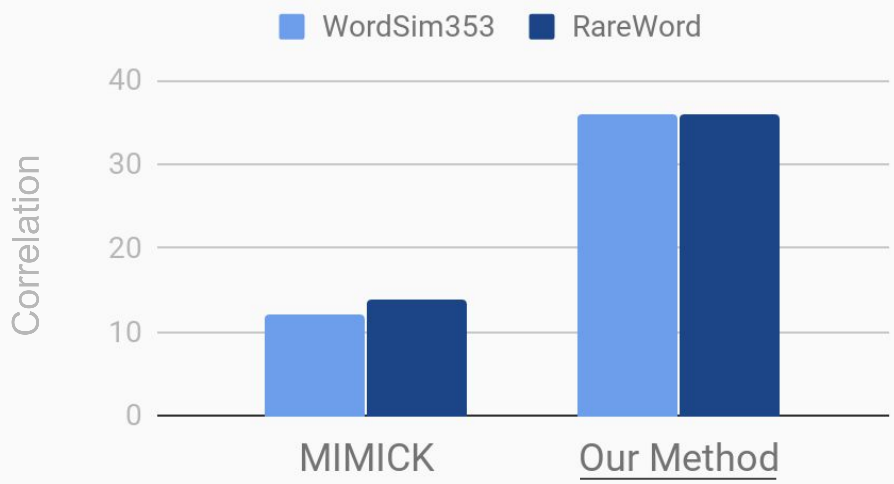
MIMICK (Pinter et al. 2017) subword-level model.

Word Similarity Task

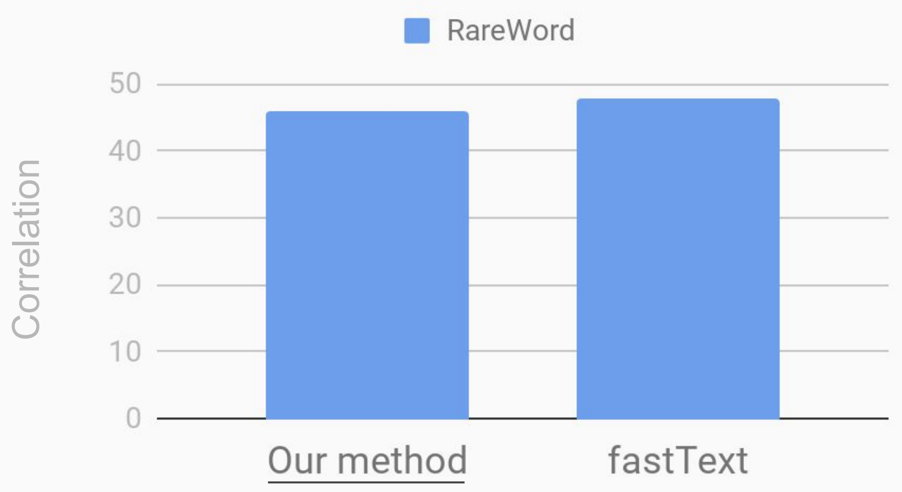
Word pairs	Human label	Induced similarity
love,sex	6.77	0.6
tiger,cat	7.35	0.5
book,paper	7.46	0.6
computer,keyboard	7.62	0.8
...		...

correlation

$cos(v_{w1}, v_{w2})$



Our method almost triples the correlation score on common and rare words compared to MIMICK.



Our method matches the performance with fastText on rare words without access to contexts.

Spelling is effective!

Word Similarity Task

Target vectors:

- English PolyGlot vectors
- Google word2vec vectors

Evaluation sets:

- RW = Stanford RareWord
- WS = WordSim353

Other approach:

- Edit distance
- fastText over Wikipedia dump

	Dim.	# Tokens	RW	WS
Polyglot	64	100k	41(58%)	45(5%)
Google	300	160k	53(11%)	69(1%)

Table 1: Target vectors statistics and word similarity task scores in Spearman's $\rho \times 100$. In parentheses are OOV rates.

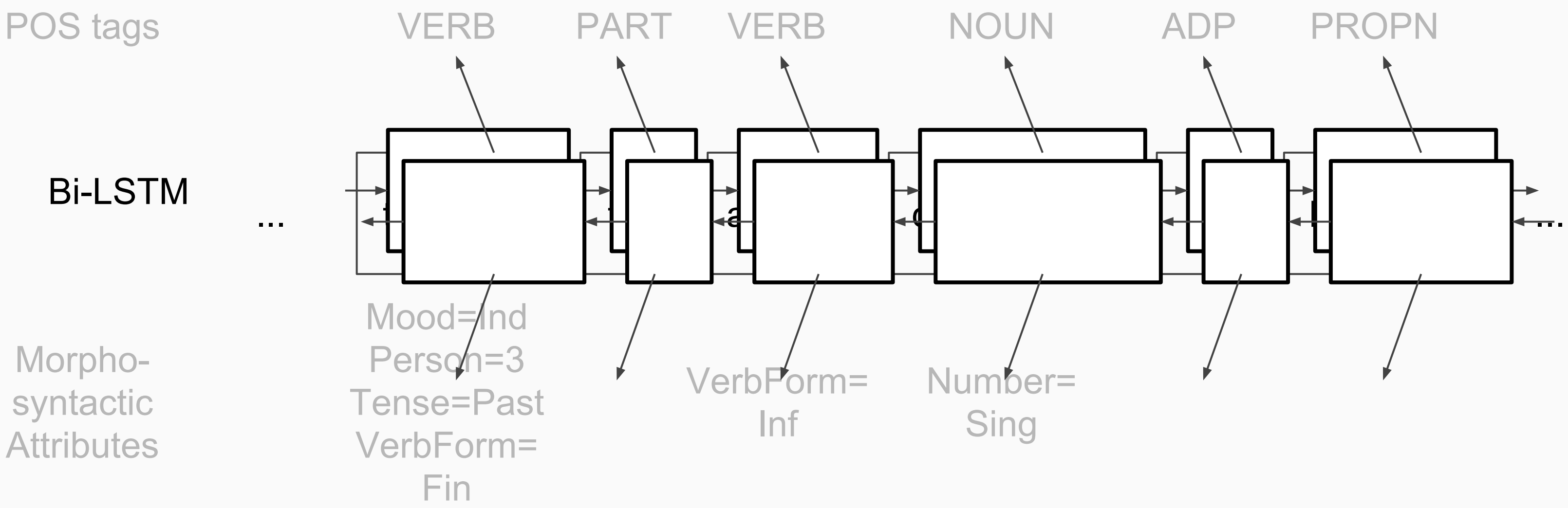
Model	Size	Target	RW	WS
EditDist	-	-	18	-2
MIMICK	649KB	Polyglot	14	12
BoS	238MB	Polyglot	36	36
BoS	1.3GB	Google	46	56
fastText	8.0GB	-	48	74

Table 2: Word similarity task results measured in Spearman's $\rho \times 100$.

Joint Prediction of Part-of-Speech Tags and Morphosyntactic Attributes

POS tags		VERB	PART	VERB	NOUN	ADP	PROPN	
Sentence	...	traveled	to	attend	conference	in	Belgium	...
Morpho-syntactic Attributes		Mood=Ind Person=1 Tense=Past VerbForm=Fin		VerbForm=Inf	Number=Sing			

Joint Prediction of Part-of-Speech Tags and Morphosyntactic Attributes



MIMICK (Pinter et al. 2017).

/ ar / bg / cs / da / el / en / es / eu / fa / he / hi / hu / id / it / kk / lv / ro / ru / sv / ta / tr / vi / zh /

23 languages

Our method **consistently outperforms** MIMICK in all the 23 languages tested within the universal dependency (UD) dataset.

	N_{train}	POS tagging			Morphosyntactic attributes		
		random	MIMICK	BoS	random	MIMICK	BoS
Kazakh	4,949	0.589	0.681	**0.758**(0.077)	0.021	0.032	**0.240**(0.208)
Tamil	6,329	0.480	0.678	**0.774**(0.097)	0.568	0.673	**0.762**(0.089)
Latvian	13,781	0.589	0.757	**0.872**(0.115)	0.374	0.572	**0.676**(0.104)
Vietnamese	31,800	0.749	0.564	**0.846**(0.282)	-	-	-
Hungarian	33,017	0.594	0.858	**0.922**(0.065)	0.569	0.775	**0.836**(0.061)
Turkish	41,748	0.636	0.767	**0.890**(0.123)	0.543	0.776	**0.826**(0.050)
Greek	47,449	0.819	0.907	**0.965**(0.058)	0.783	0.903	**0.934**(0.031)
Bulgarian	50,000	0.804	0.903	**0.971**(0.068)	0.649	0.851	**0.915**(0.064)
Swedish	66,645	0.748	0.813	**0.945**(0.132)	0.707	0.812	**0.930**(0.118)
Basque	72,974	0.662	0.823	**0.913**(0.091)	0.564	0.778	**0.820**(0.042)
Russian	79,772	0.665	0.897	**0.948**(0.051)	0.592	0.855	**0.915**(0.060)
Danish	88,980	0.788	0.834	**0.947**(0.114)	0.745	0.813	**0.927**(0.114)
Indonesian	97,531	0.724	0.788	**0.915**(0.127)	-	-	-
Chinese	98,608	0.721	0.793	**0.835**(0.042)	0.699	0.767	**0.790**(0.022)
Persian	121,064	0.843	0.866	**0.957**(0.091)	0.745	0.792	**0.918**(0.125)
Hebrew	135,496	0.814	0.858	**0.957**(0.099)	0.648	0.837	**0.903**(0.066)
Romanian	163,262	0.796	0.874	**0.956**(0.082)	0.718	0.876	**0.942**(0.066)
English	204,587	0.770	0.826	**0.932**(0.106)	0.822	0.859	**0.947**(0.089)
Arabic	225,853	0.780	0.901	**0.950**(0.049)	0.711	0.901	**0.942**(0.041)
Hindi	281,057	0.824	0.848	**0.939**(0.091)	0.863	0.888	**0.951**(0.063)
Italian	289,440	0.810	0.909	**0.964**(0.056)	0.839	0.927	**0.964**(0.037)
Spanish	382,436	0.819	0.914	**0.959**(0.045)	0.793	0.915	**0.954**(0.038)
Czech	1,173,282	0.695	0.908	**0.966**(0.058)	0.622	0.845	**0.905**(0.061)

Table 3: POS tagging accuracy and morphosyntactic attributes micro F1 over 23 languages (UD 1.4). In parentheses are the gains to MIMICK.

Efficiency

Training time.

3.5 s/epoch

Our model takes only 3.5 s/epoch to train over English PolyGlot vectors with a naive single-thread CPU-only Python implementation and a usual desktop PC.

Conclusion

A surprisingly simple and fast method to extend pre-trained word vectors towards out-of-vocabulary words, **without using any context**.

The intrinsic and extrinsic evaluations show that our model's ability in capturing lexical knowledge and generating good vectors, **using only spellings**.

Can we do more or better with spellings only or with minimal extra context?

Thanks for listening!

Q & A