

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. Many popular models to learn such representations ignore the morphology of words, by assigning a distinct vector to each word. This is a limitation, especially for morphologically rich languages with large vocabularies and many rare words. **In this paper, we propose a new approach based on the skip-gram 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 corpus quickly. We evaluate the obtained word representations on five different languages, on word similarity and analogy tasks.

1 Introduction

Learning continuous representations of words has a long history in natural language processing (Rumelhart et al., 1988). These representations are typically derived from large unlabeled corpora using co-occurrence statistics (Deerwester et al., 1990; Schütze, 1992; Lund and Burgess, 1996). A large body of work, known as distributional semantics, have studied the properties of these methods (Turney et al., 2010; Baroni and Lenci, 2010). In the neural network community, Collobert and Weston (2008) proposed to learn word embeddings using a feedforward

neural network, by predicting a word based on the two words on the left and two words on the right. More recently, Mikolov et al. (2013b) proposed simple log-bilinear models to learn continuous representations of words on very large corpora efficiently.

Most of these techniques represent each word of the vocabulary by a distinct vector, without parameters sharing. In particular, they ignore the internal structure of words, which is an important limitation for morphologically rich languages, such as Turkish or Finnish. These languages contain many words that occur rarely, making it difficult to learn good word-level representations. In this paper, we propose to learn representations for character n -grams, and 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 five different languages, with various degree of morphology, showing the benefit of our approach.

2 Related work

Morphological word representations. In recent years, many methods have been proposed to incorporate morphological information into word representations. To model rare words better, Alexandrescu and Kirchhoff (2006) introduced factored neural language models, where words are represented as sets of features. These features might include morphological information, and this technique was successfully applied to morphologically rich languages, such as Turk-

*These authors contributed equally.

ish (Sak et al., 2010). Recently, several works have proposed different composition functions to derive representations of words from morphemes (Lazaridou et al., 2013; Luong et al., 2013; Botha and Blunsom, 2014; Qiu et al., 2014). These different approaches rely on a morphological decomposition of words, while our does not. Similarly, 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. Word representations trained on morphologically annotated data were introduced by Cotterell and Schütze (2015). Closest to our approach, Schütze (1993) learned representations of character fourgrams through singular value decomposition, and derived representations for words by summing the fourgrams representations.

Character level features for NLP. Another area of research closely related to our work are character-level models for natural language processing, to learn representations directly from sequence of characters. A first class of such models are recurrent neural networks, applied to language modeling (Mikolov et al., 2012; Sutskever et al., 2011; Graves, 2013; Bojanowski et al., 2015), text normalization (Chrupała, 2014), part-of-speech tagging (Ling et al., 2015) and parsing (Ballesteros et al., 2015). Another family of models are convolutional neural networks trained on characters, which were applied to part-of-speech tagging (dos Santos and Zadrozny, 2014), sentiment analysis (dos Santos and Gatti, 2014), text classification (Zhang et al., 2015) and language modeling (Kim et al., 2016). Sperr et al. (2013) introduced a language model based on restricted Boltzmann machine, in which words are encoded as a set of character n -grams. Finally, recent works in machine translation have proposed to use subword units to obtain representations of rare words (Sennrich et al., 2016; Luong and Manning, 2016).

3 Model

In this section, we propose a model to learn representations taking into account word morphology.

General model. We start by briefly reviewing the continuous skip-gram model (Mikolov et al., 2013b) from which our model is derived. Given a vocabulary of size W , where a word is identified by its index $w \in \{1, \dots, W\}$, the goal is to learn a vectorial representation for each word w . Inspired by the distributional hypothesis (Harris, 1954), these representations are trained to predict words appearing in the context of a given word. More formally, given a large training corpus represented as a sequence of words w_1, \dots, w_T , the objective of the skip-gram model is to maximize the log-likelihood

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

where the context \mathcal{C}_t is the set of indices of words surrounding w_t . The probability of observing a context word w_c given w_t is parametrized using the word vectors. Given a scoring function s , which maps pairs of (word, context) to scores in \mathbb{R} , a possible choice to define the probability of a context word is the softmax

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

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

This problem can also be framed as a set of independent binary classification tasks, where the goal is to predict the presence (or absence) of context words. For the word at position t and the context c , we obtain the negative log-likelihood

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

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

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

A natural parametrization for the score function between a word w_t and a context word w_c is to take the scalar product between word and context embeddings $s(w_t, w_c) = \mathbf{u}_{w_t}^\top \mathbf{v}_{w_c}$, where \mathbf{u}_{w_t} and \mathbf{v}_{w_c} are vectors in \mathbb{R}^d . This is the skip-gram model with negative sampling, introduced by Mikolov et al. (2013b).

Subword model. By using a distinct vector representation for each word, the skip-gram model ignore the internal structure of words. In this section, we thus propose a different scoring function s , in order to take into account this information. Given a word w , let us denote by $\mathcal{G}_w \subset \{1, \dots, G\}$ 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

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

We always include the word w in the set of its n -grams, to also learn a vector representation for each word. The set of n -grams is thus a superset of the vocabulary. It should be noted that different vectors are assigned to a word and a n -gram sharing the same sequence of characters. For example, the word *as* and the bigram *as*, appearing in the word *paste*, will be assigned to different vectors. This simple model allows sharing the representations across words, thus allowing to learn reliable representation for rare words.

Dictionary of n -grams. The presented model is simple and leaves room for design choices in the definition of \mathcal{G}_w . In this paper, we adopt a very simple scheme: we keep all the n -grams with a length greater or equal than 3 and smaller or equal than 6. Different sets of n -grams could be used, for example prefixes and suffixes. We also add a special character for the beginning and the end of the word, thus allowing to distinguish prefixes and suffixes.

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 . In the following, we use K equal to 2 millions. In the end, a word is represented by its index in the word dictionary and the value of the hashes of its n -grams. To improve the efficiency

of our model, we do not use n -grams to represent the P most frequent words in the vocabulary. There is a trade-off in the choice of P , as smaller values imply higher computational cost but better performance. When $P = W$, our model is the skip-gram model of Mikolov et al. (2013b).

4 Experiments

Datasets and baseline. First, we compare our model to the C implementation of the skip-gram and CBOW models from the `word2vec`¹ package. The different models are trained on Wikipedia data² in five languages: English, Czech, French, German and Spanish. We randomly sampled datasets of three different sizes: small (50M tokens), medium (200M tokens) and full (the complete Wikipedia dump). All the datasets are shuffled, and we train by using five epochs.

Implementation details. For all experiments, for both our model and the baselines, we use the following parameters. We sample 5 negatives, we use a window size of 5, a rejection threshold of 10^{-4} and keep words appearing at least 5 times. On small datasets, we use vector representations of dimension 100, on medium datasets, we use vectors of dimension 200, while on the full datasets, we use a dimension of 300. The learning rate is set to 0.025 for the skip-gram baseline and to 0.05 for both our model and the CBOW baseline. Using this setting on English data, our model with character n -grams is approximately $1.5 \times$ slower to train than the skip-gram baseline (105k words/second/thread versus 145k words/second/thread for the baseline). Our model is implemented in C++, and we will make our code available upon publication.

Human similarity judgement. We first evaluate the quality of our representations by computing Spearman’s rank correlation coefficient (Spearman, 1904) between human judgement and the cosine similarity between the vector representations. For English, we use the WS353 dataset introduced by Finkelstein et al. (2001) and the rare word dataset (RW), introduced by Luong et al. (2013). For German, we com-

¹<https://code.google.com/archive/p/word2vec/>

²<https://dumps.wikimedia.org/>

	Small				Medium				Full			
	oov	sg	cbow	our	oov	sg	cbow	our	oov	sg	cbow	our
DE-GUR350	20%	52	57	64	11%	57	61	69	8%	67	69	72
DE-GUR65	0%	55	55	62	0%	64	68	73	0%	78	78	84
DE-ZG222	28%	21	29	39	18%	32	42	43	10%	37	42	44
EN-RW	45%	38	37	45	21%	37	38	46	3%	45	45	49
EN-WS353	0%	65	67	66	0%	70	72	71	0%	72	73	74
ES-WS353	1%	52	55	55	1%	57	58	57	0%	62	62	60
FR-RG65	0%	64	67	72	0%	49	56	56	0%	31	30	43
EN-SYN	1%	35	34	55	1%	54	54	65	1%	68	68	73
EN-SEM	4%	46	50	36	2%	67	71	59	1%	79	79	79
CS-ALL	24%	28	28	56	24%	46	46	62	24%	44	46	63

Table 1: Correlations between vector similarity score and human judgement on several datasets (top) and accuracies on analogy tasks (bottom) for models trained on Wikipedia. For each dataset, we evaluate models trained on several sizes of training sets. Small contains 50M tokens, Medium 200M tokens and Full is the complete Wikipedia dump. For each dataset, we report the out-of-vocabulary rate as well as the performance of our model and the skip-gram and CBOW models from Mikolov et al. (2013b).

query	tiling	tech-rich	english-born	micromanaging	eateries	dendritic
our	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 2: Nearest neighbors of rare words using our representations and skip-gram. These hand picked examples are for illustration.

pare the different models on three datasets: Gur65, Gur350 and ZG222 (Gurevych, 2005; Zesch and Gurevych, 2006). Finally, we evaluate the French word vectors on the translated dataset RG65 (Joubarne and Inkpen, 2011) and the Spanish word vectors on the dataset WS353 (Hassan and Mihalcea, 2009).³

Some words from these datasets do not appear in our training data, and thus, we cannot obtain word representation for these words using the CBOW and skip-gram baselines. Therefore, we decided to exclude pairs containing such words from the evaluation. We report the out of vocabulary rate (OOV) with our results in Table 1. It should be noted that our method and the baselines share the same vocabulary, so that results of different methods on the same training set are comparable. On the other hand, results on different training corpora are not compara-

ble, since the vocabularies are not the same (hence the different OOV rates).

First, we notice that the proposed model, which uses subword information outperforms the baselines on most datasets. We also observe that the effect of using character n -grams is significantly more important for German than for English or Spanish. This is not surprising since German is morphologically richer than English or Spanish. Unsurprisingly, the difference between our model and the baselines is also more important on smaller datasets. Second, we observe that on the English rare words dataset (RW), our approach also outperforms the baselines.

Word analogy tasks. We now evaluate our approach on word analogy questions, of the form A is to B as C is to D , where D must be predicted by the models. We use the syntactic (en-syn) and semantic (en-sem) questions introduced by Mikolov et al. (2013a) for En-

³Since words are not accented in WS353-es, we preprocess the training data similarly.

	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
Our	69	37	73	46	54	67

Table 3: Comparison of our approach with previous work incorporating morphology in word representations, on word similarity tasks. 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. We report Spearman’s rank correlation coefficient between model scores and human judgement.

glish and the dataset `cs-all`, introduced by Svoboda and Brychcin (2016), for Czech. Some questions contain words that do not appear in our training corpus, and we thus exclude these questions and report the out of vocabulary rate. All methods are trained on the same data, therefore the reported results are comparable. We report accuracy for the different models in Table 1. We observe that morphological information significantly helps for the syntactic tasks, our approach outperforming the baselines on `en-syn`. In contrast, it degrades the performance on semantic tasks for small training datasets. Second, we observe that for Czech, a morphologically rich language, using subword information strongly improves the results over the baselines.

Comparison with morphological representations.

We also compare our approach to previous work on incorporating subword information in word vectors, on word similarity tasks. The methods used are: the recursive neural network of Luong et al. (2013), the morpheme CBOW of Qiu et al. (2014) and the morphological transformations of Soricut and Och (2015). For English, we trained our model on the Wikipedia data released by Shaoul and Westbury (2010), while for German, Spanish and French, we use the news crawl data from the 2013 WMT shared task. These datasets were also used to train the models we are comparing to. Contrary to the previous experiments, we keep all the words from the evaluation data, even if they did not appear in the training data. Using our model, we obtain representations of out-of-vocabulary words by summing the representations of character n -grams. This makes our results com-

parable to those reported in previous work. We report results in Table 3. We observe that our simple approach, based on character n -grams, performs well relative to techniques based on subword information obtained from morphological segmentors. In particular, our method does not need any preprocessing of the data, making it fast to apply.

5 Discussion

In this paper, we investigate a very simple method to learn word representations taking into account subword information. Our approach, which incorporates character n -grams into the skip-gram model, is related to an old idea (Schütze, 1993), which has not received a lot of attention in the last decade. We show that our method outperforms baselines which do not take into account subword information, on rare words, morphologically rich languages and small training datasets. We will open source the implementation of our model, in order to facilitate comparison of future work on learning subword representations.

References

- [Alexandrescu and Kirchhoff2006] Andrei Alexandrescu and Katrin Kirchhoff. 2006. Factored neural language models. In *Proc. NAACL*.
- [Ballesteros et al.2015] Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. Improved transition-based parsing by modeling characters instead of words with lstms. In *Proc. EMNLP*.
- [Baroni and Lenci2010] Marco Baroni and Alessandro Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4).

- [Bojanowski et al.2015] Piotr Bojanowski, Armand Joulin, and Tomáš Mikolov. 2015. Alternative structures for character-level rnns. In *ICLR Workshop*.
- [Botha and Blunsom2014] Jan A Botha and Phil Blunsom. 2014. Compositional morphology for word representations and language modelling. In *Proc. ICML*.
- [Chen et al.2015] Xinxiong Chen, Lei Xu, Zhiyuan Liu, Maosong Sun, and Huanbo Luan. 2015. Joint learning of character and word embeddings. In *Proc. IJCAI*.
- [Chrupała2014] Grzegorz Chrupała. 2014. Normalizing tweets with edit scripts and recurrent neural embeddings. In *Proc. ACL*.
- [Collobert and Weston2008] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multi-task learning. In *Proc. ICML*.
- [Cotterell and Schütze2015] Ryan Cotterell and Hinrich Schütze. 2015. Morphological word-embeddings. In *Proc. of NAACL*.
- [Cui et al.2015] Qing Cui, Bin Gao, Jiang Bian, Siyu Qiu, Hanjun Dai, and Tie-Yan Liu. 2015. Knet: A general framework for learning word embedding using morphological knowledge. *ACM Transactions on Information Systems*.
- [Deerwester et al.1990] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6).
- [dos Santos and Gatti2014] Cicero Nogueira dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proc. COLING*.
- [dos Santos and Zadrozny2014] Cicero Nogueira dos Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *Proc. ICML*.
- [Finkelstein et al.2001] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppín. 2001. Placing search in context: The concept revisited. In *Proc. WWW*.
- [Graves2013] Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.
- [Gurevych2005] Iryna Gurevych. 2005. Using the structure of a conceptual network in computing semantic relatedness. In *Proc. IJCNLP*.
- [Harris1954] Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3).
- [Hassan and Mihalcea2009] Samer Hassan and Rada Mihalcea. 2009. Cross-lingual semantic relatedness using encyclopedic knowledge. In *Proc. EMNLP*.
- [Joubarne and Inkpen2011] Colette Joubarne and Diana Inkpen. 2011. Comparison of semantic similarity for different languages using the google n-gram corpus and second-order co-occurrence measures. In *Adv. in A.I.*
- [Kim et al.2016] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *Proc. AAAI*.
- [Lazaridou et al.2013] Angeliki Lazaridou, Marco Marelli, Roberto Zamparelli, and Marco Baroni. 2013. Compositionally derived representations of morphologically complex words in distributional semantics. In *Proc. ACL*.
- [Ling et al.2015] Wang Ling, Chris Dyer, Alan W Black, Isabel Trancoso, Ramon Fernandez, Silvio Amir, Luis Marujo, and Tiago Luis. 2015. Finding function in form: Compositional character models for open vocabulary word representation. In *Proc. EMNLP*.
- [Lund and Burgess1996] Kevin Lund and Curt Burgess. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2).
- [Luong and Manning2016] Minh-Thang Luong and Christopher D. Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. In *Proc. ACL*.
- [Luong et al.2013] Thang Luong, Richard Socher, and Christopher D Manning. 2013. Better word representations with recursive neural networks for morphology. In *Proc. CoNLL*.
- [Mikolov et al.2012] Tomáš Mikolov, Ilya Sutskever, Anoop Deoras, Hai-Son Le, Stefan Kombrink, and J Cernocky. 2012. Subword language modeling with neural networks. *preprint (http://www.fit.vutbr.cz/imikolov/rnnlm/char.pdf)*.
- [Mikolov et al.2013a] Tomáš Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- [Mikolov et al.2013b] Tomáš Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Adv. NIPS*.
- [Qiu et al.2014] Siyu Qiu, Qing Cui, Jiang Bian, Bin Gao, and Tie-Yan Liu. 2014. Co-learning of word representations and morpheme representations. In *Proc. COLING*.
- [Rumelhart et al.1988] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1988. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3).
- [Sak et al.2010] Haşim Sak, Murat Saraclar, and Tunga Gungör. 2010. Morphology-based and sub-word lan-

- guage modeling for turkish speech recognition. In *Proc. ICASSP*.
- [Schütze1992] Hinrich Schütze. 1992. Dimensions of meaning. In *Proc. Supercomputing*.
- [Schütze1993] Hinrich Schütze. 1993. Word space. In *Adv. NIPS*.
- [Sennrich et al.2016] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proc. ACL*.
- [Shaoul and Westbury2010] C. Shaoul and C. Westbury. 2010. The Westbury lab Wikipedia corpus.
- [Soricut and Och2015] Radu Soricut and Franz Och. 2015. Unsupervised morphology induction using word embeddings. In *Proc. NAACL*.
- [Spearman1904] Charles Spearman. 1904. The proof and measurement of association between two things. *American Journal of Psychology*, 15.
- [Sperr et al.2013] Henning Sperr, Jan Niehues, and Alexander Waibel. 2013. Letter n-gram-based input encoding for continuous space language models. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*.
- [Sutskever et al.2011] Ilya Sutskever, James Martens, and Geoffrey E Hinton. 2011. Generating text with recurrent neural networks. In *Proc. ICML*.
- [Svoboda and Brychcin2016] L. Svoboda and T. Brychcin. 2016. New word analogy corpus for exploring embeddings of czech words. In *Proc. CICLING*.
- [Turney et al.2010] Peter D Turney, Patrick Pantel, et al. 2010. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37(1).
- [Zesch and Gurevych2006] Torsten Zesch and Iryna Gurevych. 2006. Automatically creating datasets for measures of semantic relatedness. In *Proc. Workshop on Linguistic Distances*.
- [Zhang et al.2015] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Adv. NIPS*.