Explaining and Generalizing Skip-Gram through Exponential Family Principal Component Analysis

Ryan Cotterell **Adam Poliak** **Benjamin Van Durme** **Jason Eisner**
Center for Language and Speech Processing
Johns Hopkins University
{ryan.cotterell,azpoliak,vandurme,jason}@cs.jhu.edu

Abstract

The popular skip-gram model induces word embeddings by exploiting the signal from word-context coocurrence. We offer a new interpretation of skip-gram based on exponential family PCA—a form of matrix factorization. This makes it clear that we can extend the skip-gram method to *tensor* factorization, in order to train embeddings through richer higher-order coocurrences, e.g., triples that include positional information (to incorporate syntax) or morphological information (to share parameters across related words). We experiment on 40 languages and show that our model improves upon skip-gram.

1 Introduction

Over the past years NLP has witnessed a veritable frenzy on the topic of word embeddings: low-dimensional representations of distributional information. The embeddings, trained on extremely large text corpora such as Wikipedia and Common Crawl, are claimed to encode semantic knowledge extracted from large text corpora.

Numerous methods have been proposed—the most popular being skip-gram (Mikolov et al., 2013) and GloVe (Pennington et al., 2014)—for learning these low-dimensional embeddings from a bag of contexts associated with each word type. Natural language text, however, contains richer structure than simple context-word pairs. In this work, we embed n-tuples rather than pairs, allowing us to escape the bag-of-words assumption and encode richer linguistic structures.

As a first step, we offer a novel interpretation of the skip-gram model (Mikolov et al., 2013). We show how skip-gram can be viewed as an application of exponential-family principal components analysis (EPCA) (Collins et al., 2001) to an integer matrix of coocurrence counts. Previous work has related the negative sampling *estimator* for skip-gram model parameters to the factorization of a matrix of (shifted) positive pointwise mutual information (Levy and Goldberg, 2014b). We show the skip-gram *objective* is just EPCA factorization.

By extending EPCA factorization from matrices to tensors, we can consider higher-order cooccurrence statistics. Here we explore incorporating positional and morphological content in the model by factorizing a positional tensor and morphology tensor. The positional tensor directly incorporates word order into the model, while the morphology tensor adds word-internal information. We validate our models experimentally on 40 languages and show large gains under standard metrics.[1]

2 Matrix Factorization

In this section, we briefly explain how skip-gram is an example of EPCA. We are given data in the form of a matrix $X \in \mathbb{R}^{n_1 \times n_2}$, where X_{ij} is the number of times that word j appears in context i under some user-specified definition of "context." **Principal components analysis** (Pearson, 1901) approximates X as the product $C^\top W$ of two matrices $C \in \mathbb{R}^{d \times n_1}$ and $W \in \mathbb{R}^{d \times n_2}$, whose columns are d-dimensional vectors that embed the contexts and the words, respectively, for some user-specified $d < \min(n_1, n_2)$. Specifically, PCA minimizes[2]

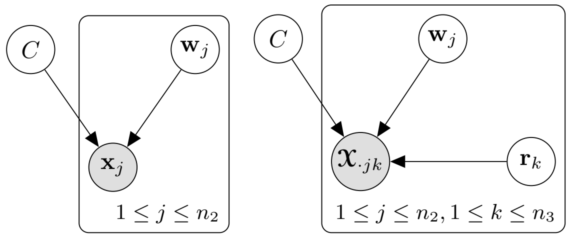
$$\left|\left|X - C^\top W\right|\right|_{\text{F}}^2 = \sum_{ij} (X_{ij} - \mathbf{c}_i \cdot \mathbf{w}_j)^2 \quad (1)$$

$$= \sum_{j} \left|\left|\mathbf{x}_j - C^\top \mathbf{w}_j\right|\right|^2 \quad (2)$$

where $\mathbf{c}_i$, $\mathbf{w}_j$, $\mathbf{x}_j$ denote the i^{th} column of C and the j^{th} columns of W and X, and $\mathbf{c}_i \cdot \mathbf{w}_j$ denotes an inner product of vectors (sometimes called "cosine

[1] The code developed is available at https://github.com/azpoliak/skip-gram-tensor.

[2] Singh and Gordon (2008) offer a comprehensive discussion of PCA and other matrix factorization techniques in ML.



(a) Matrix factorization (b) Tensor factorization

Figure 1: Comparison of the graphical model for matrix factorization (either PCA or EPCA) and 3-dimensional tensor factorization. Priors are omitted from the drawing.

similarity”). Note that $\text{rank}(C^\top W) \leq d$, whereas $\text{rank}(X) \leq \min(n_1, n_2)$. Globally optimizing equation (1) means finding the *best* approximation to X with rank $\leq d$ (Eckart and Young, 1936), and can be done by SVD (Golub and Van Loan, 2012).

By rewriting equation (1) as (2), both Roweis (1997) and Tipping and Bishop (1999) observed that the optimal values of C and W can be regarded as the maximum-likelihood parameter estimates for the Gaussian graphical model drawn in Figure 1a. This model supposes that the observed column vector $\mathbf{x}_j$ equals $C^\top \mathbf{w}_j$ plus Gaussian noise, specifically $\mathbf{x}_j \sim \mathcal{N}(C^\top \mathbf{w}_j, I)$. Equation (2) is this model's negated log-likelihood (plus a constant).[3]

However, recall that in our application, $\mathbf{x}_j$ is a vector of observed *counts* of the various contexts in which word j appeared. Its elements are always non-negative integers—so as Hofmann (1999) saw, it is peculiar to model $\mathbf{x}_j$ as having been drawn from a Gaussian. **EPCA** is a generalization of PCA, in which the observation $\mathbf{x}_j$ can be drawn according to *any* exponential-family distribution (log-linear distribution) over vectors.[4] The canonical parameter vector for this distribution is given by the j^{th} column of $C^\top W$, that is, $C^\top \mathbf{w}_j$.[5]

EPCA allows us to suppose that each $\mathbf{x}_j$ was drawn from a multinomial—a more appropriate family for drawing a count vector. Our observation is that skip-gram is precisely **multinomial EPCA with the canonical link function** (Mohamed, 2011), which generates $\mathbf{x}_j$ from a multinomial with log-linear parameterization.That is, skip-gram chooses embeddings C, W to maximize

$$\sum_j \sum_i X_{ij} \log p(\text{context } i \mid \text{word } j) \qquad (3)$$

$$= \sum_j \sum_i X_{ij} \log \frac{\exp(\mathbf{c}_i \cdot \mathbf{w}_j)}{\sum_{i'} \exp(\mathbf{c}_{i'} \cdot \mathbf{w}_j)} \qquad (4)$$

This is the log-likelihood (plus a constant) if we assume that for each word j, the context vector $\mathbf{x}_j$ was drawn from a multinomial with natural parameter vector $C^\top \mathbf{w}_j$ and count parameter $N_j = \sum_i X_{ij}$. This is the same model as in Figure 1a, but with a different conditional distribution for $\mathbf{x}_j$, and with $\mathbf{x}_j$ taking an additional observed parent N_j (which is the token count of word j).

2.1 Related work

Levy and Goldberg (2014b) also interpreted skip-gram as matrix factorization. They argued that skip-gram estimation *by negative sampling* implicitly factorizes a shifted matrix of positive empirical pointwise mutual information values. We instead regard the skip-gram objective itself as demanding EPCA-style factorization of the count matrix X: i.e., X arose stochastically from some unknown matrix of log-linear parameters (column j of X generated from parameter column j), and we seek a rank-d estimate $C^\top W$ of *that* matrix.

pLSI (Hofmann, 1999) similarly factors an unknown matrix of multinomial probabilities, which is **multinomial EPCA with the identity link function**. In contrast, our unknown matrix holds log-linear parameters—arbitrarily shifted log-probabilities, not probabilities.

Our EPCA interpretation applies equally well to the component distributions that are used in hierarchical softmax (Morin and Bengio, 2005), which is an alternative to negative sampling. Additionally, it yields avenues of future research using Bayesian (Mohamed et al., 2008) and maximum-margin (Srebro et al., 2004) extensions to EPCA.

[3] The graphical model further suggests that the $\mathbf{c}_i$ and $\mathbf{w}_j$ vectors are themselves drawn from some prior. Specifying this prior defines a MAP estimate of C and W. If we take the prior to be a spherical Gaussian with mean $\mathbf{0} \in \mathbb{R}^d$, the MAP estimate corresponds to minimizing (2) plus an L_2 regularizer, that is, a multiple of $||C||_\text{F}^2 + ||W||_\text{F}^2$. We do indeed regularize in this way throughout all our experiments, tuning the multiplier on a held-out development set. However, regularization has only minor effects with large training corpora, and is not in the original `word2vec` implementation of skip-gram.

[4] EPCA extends PCA in the same way that generalized linear models (GLMs) extend linear regression. The maximum-likelihood interpretation of linear regression supposes that the dependent variable $\mathbf{x}_j$ is a linear function C of the independent variable $\mathbf{w}_j$ plus Gaussian noise. The GLM, like EPCA, is an extension that allows other exponential-family distributions for the dependent variable $\mathbf{x}_j$. The difference is that in EPCA, the representations $\mathbf{w}_j$ are learned jointly with C.

[5] In the general form of EPCA, that column is passed through some “inverse link” function to obtain the expected feature values under the distribution, which in turn determines the canonical parameters of the distribution. We use the so-called canonical link, meaning that these two steps are inverses of each other and thus the canonical parameters are themselves a linear function of $\mathbf{w}_j$.

3 Tensor Factorization

Having seen that skip-gram is a form of matrix factorization, we can generalize it to tensors. In contrast to the matrix case, there are several distinct definitions of tensor factorization (Kolda and Bader, 2009). We focus on the polyadic decomposition (Hitchcock, 1927), which yields a satisfying generalization. The tensor analogue to PCA is **rank-d tensor approximation**, which minimizes

$$\begin{aligned} & ||\mathcal{X} - C \; {}_1\!\otimes_1 W \; {}_1\!\otimes_1 R \,||_F^2 \\ &= \sum_{ijk} (\mathcal{X}_{ijk} - \mathbf{1} \cdot (\mathbf{c}_i \odot \mathbf{w}_j \odot \mathbf{r}_k))^2 \qquad (5) \\ &= \sum_{jk} \left|\left| \mathcal{X}_{\cdot jk} - C^\top (\mathbf{w}_j \odot \mathbf{r}_k) \right|\right|^2 \qquad (6) \end{aligned}$$

Given a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, this objective tries to predict each entry as the three-way dot product of the columns $\mathbf{c}_i, \mathbf{w}_j, \mathbf{r}_k \in \mathbb{R}^d$, thus finding an approximation to $\mathcal{X}$ that factorizes into C, W, R. This polyadic decomposition of the approximating tensor can be viewed as a Tucker decomposition (Tucker, 1966) that enforces a diagonal core.

In our setting, the new matrix $R \in \mathbb{R}^{d \times n_3}$ embeds types of context-word *relations*. The tensor $\mathcal{X}$ can be regarded as a collection of $n_2 n_3$ *count vectors* $\mathcal{X}_{\cdot jk} \in \mathbb{N}^{n_1}$: the fibers of the tensor, each of which provides the context counts for some (word j, relation k) pair. Typically, $\mathcal{X}_{\cdot jk}$ counts which *context words* i are related to word j by relation k.

We now move from third-order PCA to third-order EPCA. Minimizing equation (6) corresponds to maximum-likelihood estimation of the graphical model in Figure 1b, in which each fiber of $\mathcal{X}$ is viewed as being generated from a Gaussian all at once. Our higher-order skip-gram (HOSG) replaces this Gaussian with a multinomial. Thus, HOSG attempts to maximize the log-likelihood

$$\sum_{ijk} \mathcal{X}_{ijk} \log p(\text{context } i \mid \text{word } j, \text{relation } k) \qquad (7)$$

$$= \sum_{ijk} \mathcal{X}_{ijk} \log \frac{\exp\left(\mathbf{1} \cdot (\mathbf{c}_i \odot \mathbf{w}_j \odot \mathbf{r}_k)\right)}{\sum_{i'} \exp\left(\mathbf{1} \cdot (\mathbf{c}_{i'} \odot \mathbf{w}_j \odot \mathbf{r}_k)\right)} \qquad (8)$$

Note that as before, we are taking the total count $N_{jk} = \sum_i \mathcal{X}_{ijk}$ to be observed. So while our embedding matrices must predict which words are related to word j by relation k, we are not probabilistically modeling how often word j participates in relation k in the first place (nor how often word j occurs overall). A simple and natural move in future would be to extend the generative model to predict these facts also from $\mathbf{w}_j$ and $\mathbf{r}_k$, although this weakens the pedagogical connection to EPCA.

We locally optimize the parameters of our probability model—the word, context and relation embeddings—through stochastic gradient ascent on (7). Each stochastic gradient step computes the gradient of a single summand $\mathcal{X}_{ijk} \log p(i \mid j, k)$. Unfortunately, this requires summing over n_1 contexts in the denominator of (8), which is problematic as n_1 is often very large, e.g., 10^7. Mikolov et al. (2013) offer two speedup schemes: negative sampling and hierarchical softmax. Here we apply the negative sampling approximation to HOSG; hierarchical softmax is also applicable. See Goldberg and Levy (2014) for an in-depth discussion.

HOSG is a bit slower to train than skip-gram, since $\mathcal{X}$ yields up to n_3 times as many summands as X (but $\ll n_3$ in practice, as $\mathcal{X}$ is often sparse).

4 Two Tensors for Word Embedding

As examples of useful tensors to factorize, we offer two third-order generalizations of Mikolov et al. (2013)'s context-word matrix. We are still predicting the distribution of contexts of a given word type. Our first version *increases* the number of parameters (giving more expressivity) by conditioning on additional information. Our second version *decreases* the number of parameters (giving better smoothing) by factoring the word type.

4.1 Positional Tensor

When predicting the context words in a window around a given word token, Mikolov et al. (2013) uses the same distribution to predict each of them. We propose to use different distributions at different positions in the window, via a "positional tensor": $\mathcal{X}_{\langle \texttt{dog},\texttt{ran},-2 \rangle}$ is the number of times the context word `dog` was seen two positions to the left of `ran`. We will predict this count using $p(\texttt{dog} \mid \texttt{ran}, -2)$, defined from the embeddings of the word `ran`, the position `-2`, and the context word `dog` and its competitors. For a 10-word window, we have $\mathcal{X} \in \mathbb{R}^{|V| \times |V| \times 10}$. Considering word position should improve syntactic awareness.

4.2 Compositional Morphology Tensor

For Mikolov et al. (2013), related words such as `ran` and `running` are monolithic objects that do not share parameters. We decompose each word into a lemma j and a morphological tag k. The

		ar	bg	ca	cs	da	de	el	en	es	et	eu	fa	fi	fo	fr	ga	gl	he	hi
$c = 2$	SG	.25	.22	.41	.20	.21	**.49**	.58	.44	.41	.09	.41	.39	.20	.32	**.41**	.22	.43	.31	.10
	HOSG	**.40**	**.46**	**.45**	**.36**	**.50**	.48	**.61**	**.48**	**.42**	**.28**	**.46**	**.43**	**.39**	**.40**	.40	**.29**	**.46**	**.44**	**.40**
	Δ	+.15	+.24	+.14	+.16	+.29	−.01	+.03	+.04	+.01	+.19	+.05	+.04	+.19	+.08	−.01	+.07	+.03	+.13	+.30
		hr	hu	id	it	kk	la	lv	nl	no	pl	pt	ro	ru	sl	sv	ta	tr	ug	vi
	SG	.51	.36	.41	.45	**.47**	.42	.21	.42	.30	.43	**.42**	.28	**.34**	.13	**.54**	**.60**	.22	.53	.57
	HOSG	**.53**	**.49**	.43	**.46**	.43	**.46**	**.38**	**.45**	**.47**	**.44**	**.42**	**.46**	.33	**.37**	.51	.58	**.41**	**.62**	**.60**
	Δ	+.02	+.13	+.02	+.01	−.04	+.04	+.17	+.03	+.17	+.01	0.0	+.18	−.01	+.24	−.03	−.02	+.21	+.09	+.03
$c = 5$		ar	bg	ca	cs	da	de	el	en	es	et	eu	fa	fi	fo	fr	ga	gl	he	hi
	SG	.24	.41	.39	.29	.44	.45	.54	.52	.45	.40	.40	.38	.37	.33	.39	.53	.40	.38	.48
	HOSG	**.29**	**.47**	**.42**	**.36**	**.49**	**.52**	**.60**	**.54**	**.48**	**.42**	**.45**	**.44**	**.43**	**.41**	**.42**	**.56**	**.45**	**.43**	**.51**
	Δ	+.05	+.06	+.03	+.07	+.04	+.07	+.06	+.02	+.03	+.02	+.05	+.06	+.06	+.08	+.08	+.06	+.06	+.05	+.03
		hr	hu	id	it	kk	la	lv	nl	no	pl	pt	ro	ru	sl	sv	ta	tr	ug	vi
	SG	.50	.46	.39	.42	**.47**	.43	.52	.43	.39	.41	.38	.38	.24	.40	.46	.59	.38	.57	.57
	HOSG	**.53**	**.49**	**.44**	**.50**	.40	**.46**	**.54**	**.50**	**.44**	**.47**	**.44**	**.43**	**.34**	**.46**	**.52**	**.58**	**.43**	**.63**	**.61**
	Δ	+.03	+.03	+.05	+.08	−.07	+.03	+.02	+.07	+.06	+.06	+.06	+.05	+.10	+.06	+.05	−.01	+.06	+.06	+.04

Table 1: The scores for QVEC-CCA for 40 languages. All embeddings were trained on the complete Wikipedia dump of September 2016. We measure correlation with universal POS tags from the UD treebanks.

contexts i are still full words.[6] Thus, we predict the count $\mathcal{X}_{\langle \mathtt{dog}, \text{RUN}, t\rangle}$ using $p(\mathtt{dog} \mid \text{RUN}, t)$, where t is a morphological tag such as [pos=v,tense=PAST].

Our model is essentially a version of the skip-gram method (Mikolov et al., 2013) that parameterizes the embedding of the word `ran` as a Hadamard product $w_j \odot r_k$, where w_j embeds RUN and r_k embeds tag t. This is similar to the work of Cotterell et al. (2016), who parameterized word embeddings as a sum $w_j + r_k$ of embeddings of the component morphemes.[7] Our Hadamard product embedding is in fact more general, since the additive embedding $w_j + r_k$ can be recovered as a special case—it is equal to $(w_j; \mathbf{1}) \odot (\mathbf{1}; r_k)$, which uses twice as many dimensions to embed each object.

5 Experiments

We build HOSG on top of the HYPERWORDS package. All models (both skip-gram and higher-order skip-gram) are trained for 10 epochs and use 5 negative samples. All models for §5.1 are trained on the Sept. 2016 dump of the full Wikipedia. All models for §5.2 were trained on the lemmatized and POS-tagged WaCky corpora (Baroni et al., 2009) for French, Italian, German and English (Joubarne and Inkpen, 2011; Leviant and Reichart, 2015). To ensure controlled and fair experiments, we follow Levy et al. (2015) for all preprocessing.

5.1 Experiment 1: Positional Tensor

We postulate that the positional tensor should encode richer notions of syntax than standard bag-of-words vectors. Why? Positional information allow us to differentiate between the geometry of the coocurrence, e.g., `the` is found to the left of the noun it modifies and is—more often than—close to it. Our tensor factorization model explicitly encodes this information during training.

To evaluate the vectors, we use QVEC (Tsvetkov et al., 2016), which measures Pearson's correlation between human-annotated judgements and the vectors using CCA. The QVEC metric will be higher if the vectors better correlate with the human-annotated resource. To measure the syntactic content of the vectors, we compute the correlation between our learned vector w_i for each word and its empirical distribution g_i over universal POS tags (Petrov et al., 2012) in the UD treebank (Nivre et al., 2016). g_i can be regarded as a vector on the $(|\mathcal{T}| - 1)$-dimensional simplex, where $\mathcal{T}$ is the tag set. We report results on 40 languages from the UD treebanks in Table 1, using 4-word or 10-word symmetric context windows (i.e., $c \in \{2, 5\}$). We find that for 77.5% of the languages, our positional tensor embeddings outperform the standard skip-gram approach on the QVEC metric.

We highlight again that the positional tensor exploits *no* additional annotation, but better exploits the signal found in the raw text. Of course, our HOSG method could also be used to exploit annotations if available: e.g., one would get different embeddings by defining the relations of word j to be the labeled syntactic dependency relations in which it participates (Lin and Pantel, 2001; Levy and Goldberg, 2014a).

5.2 Experiment 2: Morphology Tensor

Since the compositional morphology tensor allows us to share parameters among related word forms, we get a single embedding for each *lemma*, i.e., all the words `ran`, `run` and `running` now con-

[6] If one wanted to extend the model to decompose the context words i as well, we see at least four approaches.

[7] Cotterell et al. (2016) made two further moves that could be applied to extend the present paper. First, they allowed a word to consist of any number of (unordered) morphemes—not necessarily two—whose embeddings were combined (by summation) to get the word embedding. Second, this sum also included word-specific random noise, allowing them to learn word embeddings that deviated from compositionality.

	fr	it		de				en						
	353	353	SimL	RG-65	353	SimL	Z222	RG-65	353	MEN	MTURK	SimL	SimV	RW
SG	48.31	43.63	21.33	44.90	28.39	50.39	29.75	70.60	**64.50**	64.33	58.77	41.62	30.48	40.78
HOSG	**58.21**	**45.00**	**28.54**	**68.08**	**40.09**	**53.97**	**31.11**	**71.71**	63.72	**66.66**	**62.64**	**49.70**	**29.96**	**42.40**
Δ	+9.90	+1.37	+7.21	+23.18	+11.7	+3.58	+1.36	+1.11	-0.78	+2.33	+3.87	+8.08	+0.52	+1.62

Table 2: Word similarity results comparing the compositional morphology tensor with the standard skip-gram model. Numbers indicate Spearman's correlation coefficient ρ between human similarity judgements and the cosine distances of vectors. For each language, we compare on several sets of human judgments as listed by Faruqui et al. (2016, Table 2).

tribute signal to the embedding of `run`. We expect these lemma embeddings to be predictive of human judgments of lemma similarity.

We evaluate using standard datasets on four languages (French, Italian, German and English). Given a list of pairs of words (always lemmata), multiple native speakers judged (on a scale of 1–10) how "similar" those words are conceptually. Our model produces a similarity judgment for each pair using the cosine similarity of their lemma embeddings $\mathbf{w}_j$. Table 2 shows how well this learned judgment correlates with the average human judgment. Our model does achieve higher correlation than skip-gram word embeddings. Note we did not compare to a baseline that simply embeds lemmas rather than words (equivalent to fixing $\mathbf{r}_k = \mathbf{1}$).

6 Related Work

Tensor factorization has already found uses in a few corners of NLP research. Van de Cruys et al. (2013) applied tensor factorization to model the compositionality of subject-verb-object triples. Similarly, Hashimoto and Tsuruoka (2015) use an implicit tensor factorization method to learn embeddings for transitive verb phrases. Tensor factorization also appears in semantic-based NLP tasks. Lei et al. (2015) explicitly factorize a tensor based on feature vectors for predicting semantic roles. Chang et al. (2014) use tensor factorization to create knowledge base embeddings optimized for relation extraction. See Bouchard et al. (2015) for a large bibliography.

Other researchers have likewise attempted to escape the bag-of-words assumption in word embeddings, e.g., Yatbaz et al. (2012) incorporates morphological and orthographic features into continuous vectors; Cotterell and Schütze (2015) consider a multi-task set-up to force morphological information into embeddings; Cotterell and Schütze (2017) jointly morphologically segment and embed words; Levy and Goldberg (2014a) derive contexts based on dependency relations; PPDB (Ganitkevitch et al., 2013) employs a mixed bag of words, parts of speech, and syntax; Rastogi et al. (2015) represent word contexts, morphology, semantic frame relations, syntactic dependency relations, and multilingual bitext counts each as separate matrices, combined via GCCA; and, finally, Schwartz et al. (2016) derived embeddings based on Hearst patterns (Hearst, 1992). Ling et al. (2015) learn position-specific word embeddings (§4.1), but do not factor them as $\mathbf{w}_j \odot \mathbf{r}_k$ to share parameters (we did not compare empirically to this). As demonstrated in the experiments, our tensor factorization method enables us to include other syntactic properties besides word order, e.g. morphology. Poliak et al. (2017) also create positional word embeddings. Our research direction is orthogonal to these efforts in that we provide a general purpose procedure for all sorts of higher-order coocurrence.

7 Conclusion

We have presented an interpretation of the skip-gram model as exponential family principal components analysis—a form of matrix factorization—and, thus, related it to an older strain of work. Building on this connection, we generalized the model to the tensor case. Such higher-order skip-gram methods can incorporate more linguistic structure without sacrificing scalability, as we illustrated by making our embeddings consider word order or morphology. These methods achieved better word embeddings as evaluated by standard metrics on 40 languages.

Acknowledgements

We thank colleagues, particularly Yanif Ahmad, and anonymous reviewers for helpful discussion and feedback. The first author was supported by a DAAD Long-Term Research Grant and an NDSEG fellowship. This work was supported by the JHU Human Language Technology Center of Excellence (HLTCOE), DARPA DEFT, and NSF Grant No. 1423276. The U.S. Government is authorized to reproduce and distribute reprints for its purposes. The views and conclusions contained in this publication are those of the authors and should not be interpreted as representing official policies or endorsements of DARPA or the U.S. Government.

References

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43(3):209–226.

Guillaume Bouchard, Jason Naradowsky, Sebastian Riedel, Tim Rocktäschel, and Andreas Vlachos. 2015. Matrix and tensor factorization methods for natural language processing. In *Tutorials*, pages 16–18, Beijing, China, July. Association for Computational Linguistics.

Kai-Wei Chang, Wen-tau Yih, Bishan Yang, and Christopher Meek. 2014. Typed tensor decomposition of knowledge bases for relation extraction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1568–1579, Doha, Qatar, October. Association for Computational Linguistics.

Michael Collins, Sanjoy Dasgupta, and Robert E. Schapire. 2001. A generalization of principal components analysis to the exponential family. In *Advances in Neural Information Processing Systems 14*, pages 617–624.

Ryan Cotterell and Hinrich Schütze. 2015. Morphological word-embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1287–1292, Denver, Colorado, May–June. Association for Computational Linguistics.

Ryan Cotterell and Hinrich Schütze. 2017. Joint semantic synthesis and morphological analysis of the derived word. *CoRR*, abs/1701.00946.

Ryan Cotterell, Hinrich Schütze, and Jason Eisner. 2016. Morphological smoothing and extrapolation of word embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1651–1660, Berlin, Germany, August. Association for Computational Linguistics.

Carl Eckart and Gale Young. 1936. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218.

Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. 2016. Problems with evaluation of word embeddings using word similarity tasks. *arXiv preprint arXiv:1605.02276*.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764, Atlanta, Georgia, June. Association for Computational Linguistics.

Yoav Goldberg and Omer Levy. 2014. word2vec explained: deriving Mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.

Gene H. Golub and Charles F. Van Loan. 2012. *Matrix Computations*, volume 3. JHU Press.

Kazuma Hashimoto and Yoshimasa Tsuruoka. 2015. Learning embeddings for transitive verb disambiguation by implicit tensor factorization. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and their Compositionality*.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*, pages 539–545. Association for Computational Linguistics.

Frank L. Hitchcock. 1927. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1):164–189.

Thomas Hofmann. 1999. Probabilistic latent semantic analysis. In *Uncertainty in Artificial Intelligence*, pages 289–296.

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 *Canadian Conference on Artificial Intelligence*, pages 216–221. Springer.

Tamara Kolda and Brett Bader. 2009. Tensor decompositions and applications. *Society for Industrial and Applied Mathematics*, 51(3):455–500.

Tao Lei, Yuan Zhang, Lluís Màrquez, Alessandro Moschitti, and Regina Barzilay. 2015. High-order low-rank tensors for semantic role labeling. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1150–1160, Denver, Colorado, May–June. Association for Computational Linguistics.

Ira Leviant and Roi Reichart. 2015. Separated by an un-common language: Towards judgment language informed vector space modeling. *arXiv*.

Omer Levy and Yoav Goldberg. 2014a. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland, June. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014b. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems 27*, pages 2177–2185.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Dekang Lin and Patrick Pantel. 2001. Dirt @sbt@discovery of inference rules from text. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 323–328, New York, NY, USA. ACM.

Wang Ling, Chris Dyer, Alan W. Black, and Isabel Trancoso. 2015. Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1299–1304, Denver, Colorado, May–June. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Shakir Mohamed, Katherine A. Heller, and Zoubin Ghahramani. 2008. Bayesian exponential family PCA. In *Advances in Neural Information Processing Systems 21*, pages 1089–1096.

Shakir Mohamed. 2011. *Generalised Bayesian Matrix Factorisation Models*. Ph.D. thesis, University of Cambridge.

Frederic Morin and Yoshua Bengio. 2005. Hierarchical probabilistic neural network language model. In *Artificial Intelligence and Statistics Conference*.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *LREC*.

Karl Pearson. 1901. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 6(2):559–572.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2089–2096, Istanbul, Turkey, May. European Language Resources Association (ELRA). ACL Anthology Identifier: L12-1115.

Adam Poliak, Pushpendre Rastogi, Michael Patrick Martin, and Benjamin Van Durme. 2017. Efficient, compositional, order-sensitive n-gram embeddings. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*.

Pushpendre Rastogi, Benjamin Van Durme, and Raman Arora. 2015. Multiview LSA: Representation learning via generalized CCA. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 556–566, Denver, Colorado, May–June. Association for Computational Linguistics.

Sam T. Roweis. 1997. EM algorithms for PCA and SPCA. In *Advances in Neural Information Processing Systems 10*, pages 626–632.

Roy Schwartz, Roi Reichart, and Ari Rappoport. 2016. Symmetric patterns and coordinations: Fast and enhanced representations of verbs and adjectives. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 499–505, San Diego, California, June. Association for Computational Linguistics.

Ajit Paul Singh and Geoffrey J. Gordon. 2008. A unified view of matrix factorization models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 358–373.

Nathan Srebro, Jason D. M. Rennie, and Tommi S. Jaakkola. 2004. Maximum-margin matrix factorization. In *Advances in Neural Information Processing Systems 17*, pages 1329–1336.

Michael Tipping and Christopher Bishop. 1999. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622.

Yulia Tsvetkov, Manaal Faruqui, and Chris Dyer. 2016. Correlation-based intrinsic evaluation of word vector representations. In *RepEval*.

Ledyard R. Tucker. 1966. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31:279–311.

Tim Van de Cruys, Thierry Poibeau, and Anna Korhonen. 2013. A tensor-based factorization model of semantic compositionality. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1142–1151, Atlanta, Georgia, June. Association for Computational Linguistics.

Mehmet Ali Yatbaz, Enis Sert, and Deniz Yuret. 2012. Learning syntactic categories using paradigmatic representations of word context. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 940–951, Jeju Island, Korea, July. Association for Computational Linguistics.